(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 059 349
A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 82101013.9

(22) Date of filing: 11.02.82

(51) Int. Cl.³: G 09 G 1/16

(30) Priority: 23.02.81 US 237316
23.02.81 US 237318

(43) Date of publication of application:
08.09.82 Bulletin 82/36

(84) Designated Contracting States:
DE FR GB NL

(71) Applicant: TEXAS INSTRUMENTS INCORPORATED
13500 North Central Expressway
Dallas Texas 75265(US)

(72) Inventor: Watts, LaVaughn F.
10410 Autumn Harvest
Houston Texas 77064(US)

(72) Inventor: Smith, Ronald L.
7058 Shady Arbor
Houston Texas 77040(US)

(72) Inventor: Pandya, Yogendra C.
12810 Grand Cross
Houston Texas 77072(US)

(72) Inventor: Wood, Paul B.
9205 Statecoach
Houston Texas 77041(US)

(74) Representative: Leiser, Gottfried, Dipl.-Ing. et al,
Patentanwälte Prinz, Bunke & Partner Ernsberger Strasse 19
D-8000 München 60(DE)

(54) Display system with multiple scrolling regions.

(57) A data display and management system includes a microprocessor (10) whose functions are implemented by instructions in data from a directly connected main memory (48). A mass data storage memory (52) is connected to the main memory (48) and has permanently stored instructions set therein for the microprocessor (10). A display control system which operates asynchronously with the microprocessor includes a display controller (16), display memory (96, 104), character memory means, and a visual character attribute generator (94). By linking one or more row attribute bytes, or pointers, to each row of characters stored in the display memory (96) the display controller (16) performs character and row manipulation on a display device without transferring whole blocks of data in the display memory (96). Multi-region display segmentation into horizontal and vertical split regions, smooth or discrete scrolling of individual regions, and various editing functions are achieved by modifying the associated display memory pointers.

./...

EP 0 059 349 A2

Croydon Printing Company Ltd.

Fig. 1a

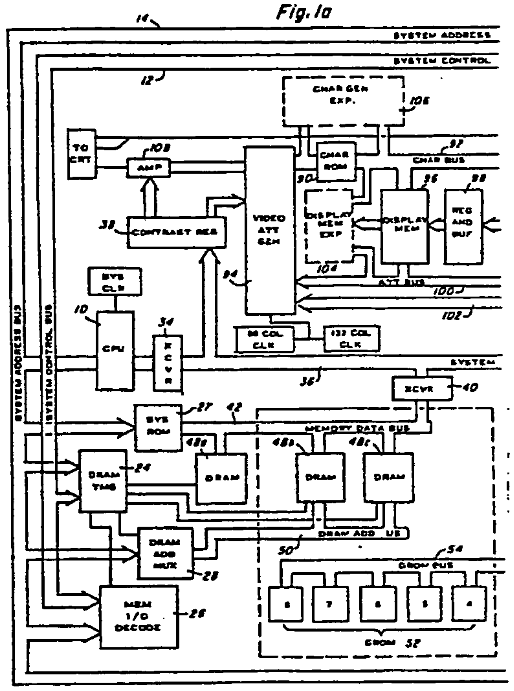


Fig. 1b

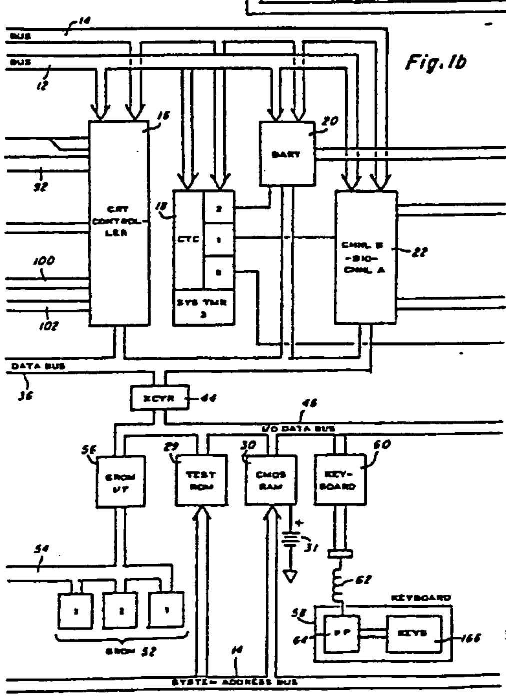
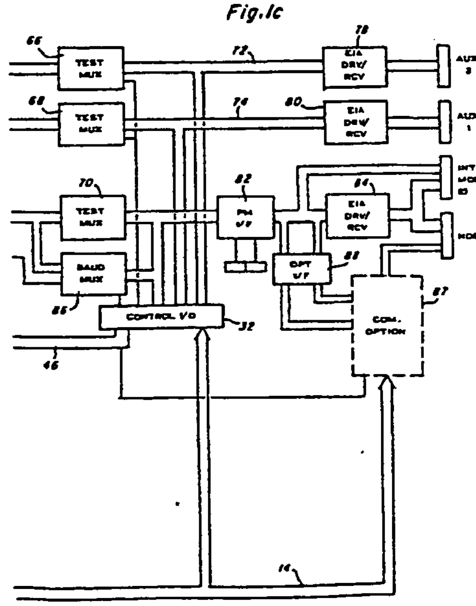


Fig. 1c



BACKGROUND OF THE INVENTION

This invention relates generally to data display and control systems and more particularly to a system having a display device segmented into multiple scrolling regions.

Traditional region separation has been performed using large scale computers. In such cases, data to be placed into a desired region was computed by software resident within the main computer. The data was then transmitted to a display device as a new display frame. Heretofore, data in display systems were not organized for display in terms of regions or windows into a data block. Further, the segmentation of the display into vertical and horizontal regions was performed by the large computer in software, and not at the local display device. Prior systems could not provide smooth or pixel scrolling of a vertical split region because of the immense software and processing time requirements.

SUMMARY OF THE INVENTION

In one embodiment of the present system, a microprocessor controls the system operation in accordance with the list of instructions stored in a random access main memory. The main memory stores a portion of the list of instructions, while the complete list of instructions is stored in a mass data storage memory in the form of one or more GROMs. An additional ROM contains a list of instructions for initializing the system and a nonvolatile memory in the form of a CMOS RAM powered by a battery to store system configuration parameters in the event of a power loss.

A display device such as a CRT is controlled by a display control system formed of a CRT controller, a visual attribute generator, a display memory, and a character ROM. Alternate character sets are provided by a character RAM which is accessed by the display controller through the attribute generator. The display controller accesses the display memory asynchronously with the system microprocessor, thus decreasing the processing time and increasing the processor throughput. Segmentation of the display into multiple horizontal and vertical split regions which are independently scrollable either discretely or smoothly, is provided by the linking of one or more row attribute bytes or "pointers" to each character

and character attribute row stored in the display memory. In this manner, data can be manipulated on the display screen without changing the absolute address of the data in display memory. That is, only the pointers associated with each block of data (corresponding to a displayed row) needs to be modified by the display controller. Processing time is greatly reduced because whole blocks of data need not be transferred in the display memory, only the display memory addresses associated with the pointers are affected.

For network applications, input/output means are provided for communicating with a controller or "host" computer. For local control of the system, input means in the form of a keyboard are provided. In one embodiment, the keyboard is a self-contained unit having a dedicated microcomputer. The keyboard unit communicates with the system through a keyboard interface unit that converts the serial data into a parallel form which is processed by the system microprocessor. Discrete control of the display device contrast is achieved by a contrast register in conjunction with the attribute generator.

It is an object of this invention to provide a display control system which permits the smooth or discrete scrolling of a plurality of horizontal and vertical split display regions.

Another object is to provide a system for controlling a display memory which minimizes the processing load on a system CPU.

Still another object is to provide a system for mapping the contents of a display memory without requiring the transfer of blocks of data therein during various editing and scroll functions.

Yet another object is to provide a display control system which may be formatted either locally or from a remote host computer.

Another object is to provide the system microprocessor with executable instructions at a high access rate, and to provide the microprocessor with instructions through a directly accessed memory whose capacity is less than the list of program instructions.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and many of the attendant advantages of the present invention will be readily apparent as the invention becomes better understood by reference to the following detailed description with the appended claims, when considered in conjunction with the accompanying drawings, wherein:

Figure 1a-1c form a block diagram of an electronic terminal including a display controller system according to the present invention;

Figure 2 is a system memory map;

Figure 3 is a diagram of the bit structure of an address byte used to access the system mass read-only program storage memory;

Figures 4 and 5 are block diagrams of the GROM interface unit;

Figure 6 is a pictorial representation showing the use of form descriptors and region data tables in establishing a display screen format;

Figure 7 shows the use of display memory mapping to scroll a selected region on a display screen segmented into a plurality of regions;

Figure 8a-8c form a block diagram of the display controller;

Figure 9 is a schematic diagram of the self-contained keyboard unit;

Figure 10 is a schematic diagram of the keyboard interface unit;

Figure 10a is a timing diagram showing the relationship of the signals in the keyboard interface unit.

DETAILED DESCRIPTION OF INVENTION

SYSTEM ARCHITECTURE

Referring now to Figures 1a-1c there is shown a block diagram of an electronic input/output terminal according to the present invention. A system microprocessor CPU 10 is coupled to a system control bus 12 and a system address bus 14. Control bus 12 and address bus 14 both extend to a CRT controller CRTC 16, a (counter-timer circuit) CTC18, a (dual asynchronous) (transmitter/receiver) DART 20, a (serial input/output device) SIO 22, a dynamic random-access memory (DRAM) timer 24, and a Memory I/O Decoder 26. System address 14 also extends to a system read-only memory (ROM) 27, a DRAM address multiplexer 28, a test ROM 29, a CMOS RAM 30, and a control I/O device 32. A battery 31 is coupled to CMOS RAM 30 to provide backup during power off conditions. CPU 10 is connected through a two-way buffer for transceiver (XCVR) 34 to a system data bus 36. Data bus 36 extends to a contrast register 38, and to CRTC 16, CTC 18, DART 20, and SIO 22. Data bus 36 is connected through a transceiver 40 to a memory data bus 42 which in turn extends to system ROM 27 and a series of DRAMs 48a-48c. Each DRAM is also connected to DRAM timer 24 and to a DRAM address bus 50 which is connected to a DRAM multiplexer 28. Data bus 36 is also connected through a XCVR 44 to an I/O data bus 46 which in turn extends to test ROM 29, CMOS RAM 30, and control I/O 32.

Program storage is provided by large scale mass read-only memory in the form of GROMs 52-1 to 52-8. The GROMs are connected to a common GROM bus 54 which is in turn connected to I/O data bus 46 through a GROM I/F 56.

An operator interfaces with the present system by means of a keyboard 58 which is connected to a keyboard I/F 60 through a telephone type multiconductor cable 62. Keyboad 58 contains its own computer 64 which permits keyboard 58 to process instructions with a minimum computing load on CPU 10.

A series of test multiplexers 66, 68, and 70, are connected respectively to DART 20, and channels B and A of SIO 22, to provide loop back testing of the terminal in a local operating mode. A set of communication buses 72, 74, and 76,

are connected to control I/O 32 and, respectively, to test multiplexers 66, 68, and 70. Communication to peripheral devices such as printers is provided from communication buses 72 and 74 through EIA driver-receivers 78 and 80, which are connected respectively to auxiliary ports AUX 2 and AUX 1. Communication with a host computer is provided by communication bus 76 which is connected through a PM interface 82 and an EIA driver-receiver 84 to an input/output port labeled HOST in the drawing. Longer distance communication is provided by means of an internal modulator-demodulator (MODEM) 85 which is also connected to communication bus 76. PM I/F 82 may be connected to a "personality module" (not shown) which permits the terminal communications to be compatible with various HOST computers. A baud rate multiplexer 86 is connected to communication bus 76 and to control I/O 32 to provide synchronization and a selection of baud rates with the HOST computer. The timing of baud multiplexer 86 is provided by timer 18-0 of CTC 18. Baud rate timing for the AUX 2 and AUX 1 ports is provided by timers 18-2 and 18-1 of CTC 18. Timer 18-3 functions as a system timer.

A communications option unit 87, shown by dashed lines in Figure 1c, is connected to the HOST port, the system address bus 14 and through an option I/F 88 to communication bus 76. Unit 87 is used to provide local and current-loop communications by methods known in the art, and need not be described in detail here.

The present terminal communicates with an operator by video display means such as a CRT (not shown) which is driven by a controller CTRC 16. The generation of characters with various character attributes will be described in more detail below. Briefly, however, a set of characters is stored in a character ROM 90 connected to CRTC 16 by a character bus 92. The character, along with selected visual "attributes" generated by a video attribute generator VAG 94 are stored in RAM display memory 96, which is connected to CRTC 16 through a register/buffer 98. In the present terminal an "attribute" is defined to mean any of the following in relation to individual characters:

high/normal intensity or background highlight,

blinking, underline, reverse video, double high characters, non-display double wide characters, and double high/wide characters. VAG 94
is connected to CRTC 16 and to display memory 96 by an attribute bus 100, and to CRTC 16 only through a control bus 102. The display memory 96 is expandable by additional RAMS 104 conected as shown in Figure 1. Additional characters may be generated by including a character generator RAM 106 connected to character bus 92 and to VAG 94 through a dot bus 107. The CRT intensity is controlled by programmable contrast register 38 through a video amplifier 108, which is also connected to VAG 94.

SYSTEM MEMORY MAP AND ARCHITECTURE

Referring now to Figure 2, there is shown a memory map of the present system illustrating the assigned memory of the various memory devices in hexidecimal form. In the basi 8-bit bytes in memory locations 0000-7FFF, and is expandable to 64K bytes in memory locations 8000-FFFF by the addition of DRAMs 48b and 48c.

Memory locations 0000-1FFF are assigned to system ROM 27, which contains 2K bytes (expandable to 8K) of program instructions to control the system initialization and diagnostics on power up, as well as instructions for loading DRAM 48a from the program storage memory resident in the GROMs. DRAM 48a is used by CPU 10 for execution of instructions to control the terminal.

Locations 2,000 - 3,000 are assigned to test ROM 29 (4K byte maximum) which is used primarily during manufacture for test and diagnostic aids such as signature analysis.

Address 3,000 is a write-only memory location that stores data in contrast register 38 to control the intensity level of the CRT as set by the operator. The attributes video on/off, background select, and highlight select are also controlled by register 38.

A read at memory location 3A00 inputs the data sent by keyboard 58 and resets its associated interrupt. A write to this location sends data to the keyboard for processing.

Read and write commands to memory locations 3A80-3A87 communicate the terminal configuration from control I/O 32 to

CPU 10, that is, which of the basic and expandable devices are installed, as well as configuring the operational mode of the various communication devices as selected by the operator. Location 3B00 is reserved for communication option unit 87, which functions as described above.

Locations 3E00-3EFF are assigned to the 256 byte CMOS RAM 30, which is powered by a battery 31 to preserve its contents during power off conditions. CMOS RAM 30 stores terminal configuration information such as communications baud rates so that the terminal can readily communicate with the HOST computer when power is restored.

Display memory 96, which stores characters and associated attributes for display on the CRT screen, is accessed through CRT controller 16 as an I/O device. Display memory 96 contains 2K 16-bit bytes and is expandable to 8K in 2K increments.

The GROM storage memory is accessed by addressing the GROM I/F 56 at memory location 3B80. A read from this location inputs 4 bits of data from the selected GROM, and a write to this memory location outputs address and control information for accessing the GROM. In the basic terminal configuration 3 16K byte GROMs, 52-1, 52-2, and 52-3, form a 48K byte program storage memory. This is expandable to a total of 128K bytes in 16K increments by the addition of GROMs 52-4 to 52-8. Dashed line 110 in Figure 1a denotes the expanded portions of the operating and program storage memory. Basically, a GROM is a PMOS, metal gate read-only memory that provides low cost mass data storage. In the present terminal, program instructions are down-loaded from the GROM to the DRAM operating system memory as needed for execution by CPU 10. In this manner, a 16K DRAM can be used for execution of 48K of program instruction. A similar method of overlaying program instructions into the DRAM is described in a co-pending patent application Serial No. 191,892 by Skelton, et al entitled "Virtual Memory Microcomputer Architecture."

Referring now to Figures 3-5, the GROM accessing procedure is shown in greater detail. Figure 3 shows the bit structure of a 20-bit byte used to access the GROM. The byte is loaded into GROM I/F 56 in 4-bit blocks or "nibbles" N1-N5.

Bits 0-13 are used to select any one of 16K byte addresses in a particular GROM, while bits 14-17 are used to select one of up to 8 GROMS. One embodiment of the GROM I/F is shown in Figure 4 and 5, illustrating hardware for accessing a 16K byte GROM through a 4 bit data/address line D0-D3.

DISPLAY CONTROL SYSTEM

The display control system of the present terminal is formed of CRT controller 16, video attribute generator 94, display memory 96, character ROM 90, contrast register 38 and the associated circuitry and expandable memories 104 and 106. Briefly, CRTC 16 provides horizontal and vertical timing for the CRT, addresses display memory 96 for CRT refresh, provides the scan line addresses for character ROM 90 and character generator 106, controls the cursor on the CRT, controls data transfers between CPU 10 and display memory 96 or character generator 106, and controls video attribute generator VAG 94. VAG 94 converts parallel dot data into serial video data, modifies dot data according to attributes and control inputs, generates a character clock from a video dot clock, and provides a data path for character generator access. The CRT is configured to display information in a 25 row by 80 or 132 column format, with the 25th line reserved for terminal status messages. Each character cell defined by a row/column address is 9 x 11 dots or pixels, with a character size of 7 x 9 pixels.

The present terminal, as configured by either the operator or the HOST computer, allows for dividing the display memory into partitions. Each partition is called an editing boundary. The partition is defined in terms of rows of display, for each row is 80 or 132 columns of display information. Additionally, the editing boundary may be defined to be an unformatted display containing no fields, but containing display attributes on a character-by-character basis. The formatted display, called a FORM, contains at least one field and may cotain display attributes on a character-by-character basis.

Only one editing boundary is displayed at a time. If more than one editing boundary is resident in the terminal, the length of each editing boundary must be equal to or greater

than 24 rows of display information. That is, if 12 rows of double high/wide data are used, then the 12 rows of double high/wide are 24 rows of display information.

A formatted editing boundary consists of "regions." A region is defined as a consecutive number of display rows viewed by the operator. The maximum number of consecutive rows possible for a region is 24 and the minimum is 1. However, the region data may exceed the display rows allocated for the defined region. This is called a scrolled region. Additionally, a region is defined as a full partition, left partition, or right partition. A full partition region is a region displayed to the operator using all columns possible in the CRT display. A left partition region is a region displayed to the operator using all columns to the left of the vertical split defined by the FORM. A right partition is a region displayed to the operator using all columns from and including the "vertical split" to the right defined by the FORM. If a region is defined as display only, with no operator access to the region, an entire region may be defined a as "protected region." A protected region is always guarded for transmission. A vertical split may exist for any FORM, however, only one vertical split may exist for any one form. A vertical split is a selected column 1-80 or 1-132 which defines the starting column of any right partition region.

Each region is a "window." A window which shows all the data in the display at once is called a fixed region or a non-scrollable region. A window that shows only a part or a segment of the data assigned to the window is called a scrolled region. Every editing boundary maintains at least one window. The window may consist of all of the editing boundary which in turn may consist of all terminal memory, but the window still exists. Each region can be individually scrolled either discreetly or smoothly. Thus, in effect, each region can function as an independent display.

These and other features are obtained by means of a "data-linking" procedure whereby two "line attributes" are generated and stored in display memory 96 for each row of characters and character attributes on the CRT screen. A line attribute is a byte of data which functions as a pointer,

telling CRT controller 16 at which display memory location the first character in the next row of characters appears. In the case of a vertical split region, wherein a full partition region is followed by a left and a right partition region on the display, the last character of the last row of the full partition region will be followed in display memory by a right attribute (RA) and a left attribute (LA). The LA points to the display memory location of the first character in the first row in the left partition, and the RA points to the memory location of the first character in the first row of the right partition. In this manner, various display functions such as inserting/deleting lines and scrolling are achieved by merely modifying the pointers RA and LA to indicate the memory location of the next desired row of characters. This greatly increases the processing throughput of CPU 10 because only the affected pointers need to be modified, without rearranging character and character attribute data in the display memory.

By way of example, and to aid in the understanding of the form and region concept described above, reference is made to Figure 6 wherein the display screen as seen by an operator is partitioned into four regions. It is assumed for this example that the terminal has been configured by the host computer to define two editing boundaries or forms. The form descriptors as well as region data tables (RDT) are loaded into the terminal memory or DRAM. Form descriptor 2 contains the information for configuring the display screen as shown in the drawing, that is, there is a vertical split at column 40 of the display screen and the form starts at display memory location 10 and ends at display memory location 50. As shown in the display memory map each of the four regions is a scrollable region, that is, the number of rows in each region exceeds the available "window" for display. Since the terminal in this example is configured to have a two page display memory, there are 48 lines of information available for the region definitions, although only 24 lines may be displayed at one time. Form 2 has four associated region data tables (RDT), one for each region on the display screen. Each RDT contains 8 bytes of information. Byte 0 is a predetermined region number, for example, to indicate whether a left or a right partition

region will be displayed on the screen. Byte 1 indicates the starting row on the display screen for the associated region. Byte 2 is the current memory display starting row, that is, the display memory address of the currently displayed row. Byte 3 indicates the memory starting row number in the region, and byte 4 is the display ending row. Byte 5 is the current memory display ending row, byte 6 is the region memory ending row number in display memory, and byte 7 is the maximum relative column number for the region on the display screen. For example, referring to the RDT for reigon 1 in Figure 6, region 1 character and character attribute data are stored in display memory locations 10-24. The currently displayed information from region 1 on the display screen, however, is formed of 8 lines that are stored in display memory locations 12-19, and appear on the display screen rows 0-7. That is, memory locations 12-19 comprise the "window" of region 1 that is currently displayed. Since region 2 is a right partition split region occupying the same display screen rows as region 1, the display window for region 2 is the same size as the window for region 1. It should be noted that 48 rows of display memory have been used for form 2, although only 24 rows of information appear on the display screen at a time. Regions 3 and 4 are similarly defined by their associated RDTs.

Although in Figure 6 the rows of information in display memory corresponding to the rows on the display screen are shown in sequential order, they may in fact be located at any display memory location. The RA and LA pointers associated with each row of screen information , as described above, will map the location of the next characters in display memory to be displayed on the screen.

Referring now to Figure 7, the use of these pointers in scrolling and individual region on the display screen will now be described by way of example. The display screen is shown divided into four regions, with the letters in each region indicating the first and last rows of characters in that particular region, for example, line B in region 1 is the first row of characters and line C represents the last row of characters in region 1. Similarly, line J represents the first row of characters in region 3 while line K represents the last row of characters. The "after" view of the display screen show region 3 scrolled up one row of characters, that is, row K is now the first displayed row of characters and row L is the last displayed row. Also shown in FIGURE 7 is a map of the RA and LA pointers showing the action of the pointers both before and after the scroll function has been completed. Before scrolling is initiated pointer LA of the status line points to row B in region 1 and pointer LA of row B points to row C. Since region 2 is a left partition split region, pointer LA of row C points to row F in region 2, while pointer RA of row C points to row J in right partition region 3. Pointer LA of row G in region 2 points to row N in region 4. Because region 3 is a right partition region, only the associated RA pointers are used to indicate the next successive row in that region. Region 3 is scrolled up by one row merely by changing pointer RA in row C of region 1 so that it points to row C of region 1 so that it points to row K in region 3 instead of row J. Thus, scrolling and other editing functions are achieved by changing the pointers only, rather than rearranging whole blocks of data in display memory as was done in prior art systems. Further, the use of controller 16 and its associated circuitry to assign and modify the various pointers greatly reduces the processing time required to perform these functions and, since CRTC 16 operates

asynchronously with CPU 10, the processing load on CPU 10 is reduced.

Smooth scrolling, i.e., the scrolling of a line one pixel or scan line at a time, may be performed independently in any of the vertical or horizontal split regions under the control of CRTC 16, which will now be described in greater detail.

TABLE 1

SELECT LINE (A0-A2) WRITE ASSIGNMENTS

$\overline{CE}$	$\overline{WR}$	A2	A1	A0	REGISTER LOADED
1	X	X	X	X	Address decoding inhibited
0	0	0	0	0	Control Reg.
0	0	0	0	1	Command Reg.
0	0	0	1	0	Character Data Reg.
0	0	0	1	1	Attribute Data Reg.
0	0	1	0	0	Cursor Address Reg. (Upper 6 bits)
0	0	1	0	1	Cursor Address Reg. (Lower 8 bits)
0	0	1	1	0	Vertical Split Reg.
0	0	1	1	1	Not used

TABLE 2

SELECT LINES (A0-A2) READ ASSIGNMENTS

$\overline{CE}$	$\overline{RD}$	A2	A1	A0	REGISTER READ
1	X	X	X	X	Address decoding inhibited
0	0	0	0	0	Status Reg.
0	0	0	0	1	Not used
0	0	0	1	0	Character Data Reg.
0	0	0	1	1	Attribute Data Reg.
0	0	1	0	0	Cursor Address Reg. (Upper 5 bits)
0	0	1	0	1	Cursor Address Reg. (Lower 8 bits)
0	0	1	1	0	Not used
0	0	1	1	1	Not used

TABLE 3

CONTROL REGISTER BIT ASSIGNMENTS

CONTROL BIT	NAME	DESCRIPTION
7		Not used
6		Not used
5	TSTMD	Test mode
4	COUT	Control out
3	ENLA	Enable row attributes
2	CBLINK	Blink cursor
1	CURON	Enable cursor
0	DFMS	Select alternate display format

DISPLAY CONTROLLER

In one embodiment, display controller CRTC 16 is a single integrated circuit device, which generates timing signals for a standard or a nonstandard raster-scan CRT monitor incorporating a non-interlaced format. CRTC 16 controls the horizontal and vertical display formatting, the independent display refresh memory 96, and the cursor address. Internal character and attribute data registers sychronize attribute and character data allowing controlled transfers between CPU 10 and display memory 96 or video attribute generator 94. As described above, controller 16 and its associated circuitry provide smooth scrolling and split screen capabilities as well as determining the number of characters per row, the number of scan lines per row, and the number of rows per display frame. In conjunction with VAG 94, the various character attribute features are also provided.

Referring now to Figure 8a-8c, CRTC 16 is shown in block diagram form. Controller 16 interfaces with CPU 10 through system address bus 14 (select lines AO-A2), system data bus 36 (data lines D0-D7), and system control bus 12 (RD/, WR/, and CE/). Lines A0-A2 are connected to an address decode logic circuit 112. Lines D0-D7 are connected to a data buffer 113 which is in turn connected to an internal data bus 114. The signal lines from control bus 12 are connected to an I/O control logic circuit 116. Address decoder 112 controls the various internal registers connected to data bus 114 by decoding the three address inputs together with the I/O control logic 116 outputs. Tables 1 and 2 below list the write and read assignments for the three select lines, A0-A2. I/O control 116 gates the control signals from CPU.
10 and controls data buffer 113 and the bus precharge.

A control register 118 connected to data bus 114 is a write-only register that contains the control bits listed in Table 3. Bit 0 (DFMS) selects either a standard or an alternate display format. The logical value of bit 0 is output as the ADF signal shown in Figure 8. Bit 1 (CURON) enables the cursor output. Setting control bit 1 to zero inhibits the cursor output unconditionally. Bit 2 (CBLINK) selects either a blinking or nonblinking cursor. Bit 3 (ENLA) sets the mode of

operation for controller 16. Enabling the line attributes allows for a display memory format with row attributes following each row. When disabled, the display format is assumed to be conventional, that is, each row is sequentially followed by the next displayed row with no row attributes present. The logical value of control bit 4 (COUT) appears as signal (COUT/) during a read or write to display memory 96. During active display this signal is inactive (High) and during retrace the signal is active (Low). Setting control bit 5 (TSTMD) reduces the display format to 16 characters per row and 2 scan lines per character row. Resetting this bit enables normal device operation. This is primarily used for testing and inspection purposes.

A command register 120 connected to data bus 114 is an eight-bit write-only register which allows CPU 10 to issue display memory read or write commands, scroll commands, or interrupt commands. The register bit assignment is shown in Table 5 wherein an "x" indicates a "don't care" value.

A character register 122 connected to data bus 114 is an eight-bit read/write register used by CPU 10 for transferring data to and from display memory 96. The lower seven bits from register 122 appear on an internaal data bus 122a. Bus 122a is connected to a character data buffer 123 which is in turn connected to character data bus 92. An attribute register 124 is a eight-bit read/write register similar to character register 122. Register 124 is connected to an internal attribute data bus 124a which is in turn connected to an attribute data buffer 125. The eight attribute bits and the most significant character bit are connected to external attribute bus 100 as shown in Figures 1a and 1b. A cursor address register 126 connected to data bus 114 is a thirteen-bit read/write register containing the absolute display memory address of the displayable cursor. The contents of this register are also the address used for display memory transfers.

A status register 128 connected to data bus 114 is a 6-bit read-only register with a bit assignment as shown in Table 4. Bit 0 (SVBLK) indicates the start of the vertical blanking interval, i.e., the beginning of the status row. Bit

TABLE 4
STATUS REGISTER BIT ASSIGNMENTS

STATUS BIT	NAME	DESCRIPTION
7		Not used
6		Not used
5	SCRBSY	Scroll busy
4	RWBSY	Read/write busy
3	INTP	Interrupt pending
2	SCRCOM	Scroll complete
1	RWCOM	Read/write complete
0	SVBLK	Vertical blank

TABLE 5
COMMAND REGISTER BIT ASSIGNMENT

COMMAND REG. BIT								DESCRIPTION
7	6	5	4	3	2	1	0	
								READ/WRITE COMMAND
0	0	X	0	0	X	0	X	No operation
						1	0	Read at cursor address
						1	1	Write at cursor address
0	0	X	0	1	0	1	0	Read and post-decrement at cursor address
				1	1	1	0	Read and post-increment at cursor address
				1	0	1	1	Write and post-decrement at cursor address
				1	1	1	1	Write and post-increment at cursor address
0	0	X	0	1	0	0	X	Enable post-decrement at cursor address
				1	1	0	X	Enable post-increment at cursor address
0	0	X	1	0	X	0	X	Post increment/decrement at cursor address
7	6	5	4	3	2	1	0	SCROLL COMMAND
0	1	X	X	X			0	Scroll up
							1	Scroll down
						0		Offset counter internal clock mode
						1		Offset counter external clock mode
					0			No operation
					1			Offset counter external increment clock
7	6	5	4	3	2	1	0	INTERRUPT CONTROL COMMAND
1	0	X					0	Disable vertical blank interrupt mask
							1	Enable vertical blank interrupt mask
						0		Disable read/write interrupt mask
						1		Enable read/write interrupt mask
					0			Disable scroll interrupt mask
					1			Enable scroll interrupt mask
				0				No operation
				1				Reset interrupts
			0					Disable interrupts
			1					Enable interrupts
1	1	X	X	X	X	X	X	No operation

1 (RWCOM) indicates that a read or write operation to display memory 96 has been completed. Bit 2 (SCRCOM) indicates that a smooth scroll operation has been completed. This will occur at the beginning of the status row on the last frame of the scroll. Bit 3 (INTP) indicates that an interrupt condition has occurred. This bit is reset or enabled by means of the command register 120. Bit 4 (RWBSY) is set following a read or a write command from CPU 10, indicating that the desired operation has not been completed. Upon completion of the read/write, this bit is reset. Bit 5 (SCRBSY) is set following a smooth scroll command from CPU 10 and is reset upon receipt of a scroll complete signal.

A vertical split register 130, connected to data bus 114, an 8-bit write-only register containing the column position of a vertical split and is zero based. That is, if register 130 is loaded with zero, on a split region the left partition would contain one character and the right partition would contain the remaining number of characters that are displayable.

The overall display format is controlled by a series of cascaded free-running counters. Two independent formats are masked programmable, each having two masked programmable formats for providing separate refresh rates. A horizontal sync register 132 is a 8-bit counter that controls horizontal display timing and is incremented at the character rate by a CCLK signal from video attribute generator 94. A programmable logic array (PLA) decodes the current register value and outputs signals corresponding to the total number of displayed characters, the start of horizontal sync, the end of horizontal sync and the end of horizontal retrace which clears the register. The operation of PLA's are known in the art and need not be described in detail. A scan line/vertical adjust counter 134 is a 4-bit counter that is incremented at the end of horizontal retrace, thus counting the number of scan lines per character row. Its count is decoded by means of a PLA and when the total number of scan lines is reached it is cleared and the counter in a vertical sync register 136 is incremented. When the register 136 counter reaches its total number of character rows count, register 134 goes into a vertical adjust

mode allowing the resolution of the refresh rate to within one scan line. Upon completion of vertical adjust, register 134 is cleared and begins counting scan lines in the upper-most character row.

Vertical sync register 136 as described above counts character rows and is clocked by register 134. Register 136 is a 7-bit counter which is PLA decoded for total displayed character rows, status row position, total character rows (including vertical retrace), and vertical sync position. The vertical sync pulse is set at vertical sync position and reset on the total character rows count.

An address counter 138 is a 4-bit counter that counts character scan lines as does register 136. Register 136 is used for normal characters while address counter 138 is used for double high and smooth scroll characters. Counter 138 tracks register 136 until one of these special character regions are detected. For double high characters counter 138 toggles between increment and inhibit increment states. For the scrolling function, counter 138 is preset with an offset signal from an offset counter 140 and then is incremented as before. Offset counter 140 is a four-bit counter which is incremented every time the status row position is reached, or once every frame. This increases the smooth scrolled offset one pixel (or scan line per region) per display frame. Offset counter 140 is initialized when a smooth scroll command is received from CPU 10 by means of command register 120. This either presets or clears counter 140 based upon the direction of the scroll. A blink rate register 142 is a three stage ripple counter that is clocked by the carry output of the most significant stage of offset counter 140. It is used to establish the cursor blink rate and the blink rate output.

The display memory 96 is addressed by the thirteen bit display memory address (DMA) outputs (DMA0-DMA12), from a buffer 143 connected to an internal DMA bus 143a. Associated control signals include write control (DMWR/), select memory or character generator (CGA), and bank select control (COUT). A display memory address counter 144 generates the DMA for active, or visible, display. Counter 144 is connected to bus 143a and contains three registers: a region 1 register 146, a

region 2 register 148, and a split register 150. Register 146 contains the starting address of a character row and is loaded into the memory address counter 144 at the beginning of every scan line of that particular row. After the last scan line of a row has been displayed, register 146 gets updated with the starting address of the next row. This may be done either by: loading memory address counter 144 contents plus 1 into register 146 when scan line attributes are disabled; loading the address following the row attribute in register 146 when the link address is not valid and row attributes are enabled; or, loading the thirteen bit address stored in memory 96 at the row attribute location into register 146, when the link address is valid and row attributes are enabled.

Region 2 register 48 is the equivalent of register 146 for a split region. That is, register 146 contains the starting address for the left region, while register 148 contains the starting address for the right region. Register 148 is loaded from either the memory address counter 144 or from display memory 96 through the internal attribute and character data buses 124a and 122a.

Split register 150 is loaded from memory address counter 144 on a split row with the display memory address of the row attribute of the left region. During the retrace state times, register 150 is loaded into the display address counter 144 to fetch data at that location. When not on the last scan line of a split row, register 150 contains a refresh address that is loaded into memory address counter 144 following the state time throughout horizontal retrace for the purpose of refreshing dynamic memory devices. When a read or a write to display memory command is given to the controller 16, it is synchronized in command register 120 and decoded. During the actual read or write operation, which occurs during the state time of retrace, the contents of cursor address register 126 are placed onto display memory address bus 143a. If the cursor is enabled, the cursor output will be active since a cursor address register comparator 152 always compares the contents of register 126 with what is on the DMA bus 123a. In this case it is comparing the register 126 contents with itself.

The character scan line outputs CSL0-3 from a scan line control logic circuit 154 may also be considered to be a part of the display memory address. These outputs indicate which scan line of the character the display memory address corresponds to. The four bits may be either from counter 138, counter 134 or the four least significant bits of the cursor address register 126. The selection between these counters depends upon the type of row being displayed, i.e., normal, smooth scrolled, or double high. The four cursor bits are always selected during the state times and are used in accessing the external character generator RAM 106. A PLA follows a multiplexer in circuit 154 for the purpose of decoding the underline position based on the selection of a standard or an alternate display format.

A state timing logic circuit 156 is a seven bit shift register with recirculation capability. At the beginning of horizontal retrace, a one is shifted into this register and serially propagates through all stages. The outputs of the stages represents state times which are used to control a register control logic circuit 158. All row attribute fetches and display memory reads and writes occur during the state times with the row attribute fetch having priority over the command oriented memory accesses. The register control logic embodies control of the character 122 and attribute 124 registers with respect to the character 92 and attribute 100 external data buses. Also controlled by logic circuit 158 are the register transfer of memory address counter 144, the command register 120 sychronization and control logic, the precharging of character bus 122a and attribute bus 124a, and the control of buffers 123 and 125. Following the last displayed character row until the beginning of vertical adjust, the state timer recirculates shifting the "one" from the last stage back to the first, thereby increasing the bandwidth of the processor (CPU 10) access to display memory 96.

A row attribute logic circuit 160 provides an input to the register control logic 158 based upon the row currently being displayed and the row to be display next. The outputs of the row attribute logic 160 provide for the following:

controlling counter 138 by presetting it with the offset counter 140, by clearing it, or by enabling it to count double high; telling the register control logic 158 when a left and/or a right row attribute fetch should occur; and determining when register 146 or register 148 should be loaded from the character and attribute buses 122a, 124a. A set of memory control buffers 162 is connected to row attribute logic 160 and to register control logic 158 in order to buffer the inputs and outputs to display memory 96 and VAG 94.

Table 6 below summarizes the various signals and I/O ports on controller 16. The registers, PLA's and counters described above are known in the art and the electronics circuitry of each need not be described in detail.

TABLE 6

SIGNATURE	PIN	I/O	DESCRIPTION
D7 (MSB)		I/O	- D7 through D0 constitutes the data bus. The data bus serves as an input for register contents to the controller and output for buffer/cursor address read data and attribute/character data.
D6		I/O	
D5		I/O	
D4		I/O	
D3		I/O	
D2		I/O	
D1		I/O	
D0 (LSB)		I/O	
A2 (MSB)		I	- A2 through A0 are select lines, used to select the internal registers of the controller.
A1		I	
A0 (LSB)		I	
$\overline{RD}$		I	- Read - a negative true signal indicates that the controller should gate data onto the data bus.
$\overline{WR}$		I	- Write - a negative true signal indicates that the controller should gate data bus into internal register.
$\overline{CE}$		I	- Chip Enable - a negative true signal indicates that the controller is being addressed.
$\overline{RESET}$		I	- Reset - a negative true signal initializes the controller, and is active for 1 µs minimum.
$\overline{INT}$		O	- Interrupt - a negative true signal indicates that at lease one of the interrupt conditions has occured. $\overline{INT}$ is an open drain output.
50HZ		I	- 50/60 Hz Control Input - this input indicates to the controller to switch to second set of vertical timing parameters in the current display format.
HSYNC		O	- Horizontal Sync - a positive true signal is used to drive the horizontal deflection circuits of a CRT.
$\overline{VSYNC}$		O	- Vertical Sync - a negative true signal is used to drive the vertical deflection circuits of a CRT.
CBLANK		O	- Composite Blanking - a positive true signal is used to blank the video input of a CRT during Horizontal and Vertical Retrace.
DMA12 (MSB)		O	- DMA12 through DMA0 constitutes the Display Memory Address. The controller outputs the address of the Display Memory location which it is currently accessing.
DMA11		O	
DMA10		O	
DMA9		O	
DMA8		O	
DMA7		O	
DMA6		O	
DMA5		O	
DMA4		O	
DMA3		O	
DMA2		O	
DMA1		O	
DMA0 (LSB)		O	

TABLE 6 (cont'd)

SIGNATURE	PIN	I/O	DESCRIPTION
VCC 1		I	- Supply Voltage (+5V NOM).
VCC 2		I	- Supply Voltage (+5V NOM).
VSS		I	- Ground Reference Voltage.
CD6 (MSB)		I/O	- CD6 through CD0 constitutes the character data bu This data bus defines the character the controlle stores or reads from the Display Memory.
CD5		I/O	
CD4		I/O	
CD3		I/O	
CD2		I/O	
CD1		I/O	
CD0 (LSB)		I/O	
AD8 (MSB)		I/O	- AD8 through AD0 constitutes the Attribute Data Bus This data bus defines the Visual Attributes the co troller stores or reads from the Display/Memory.
AD7		I/O	
AD6		I/O	
AD5		I/O	
AD4		I/O	
AD3		I/O	
AD2		I/O	
AD1		I/O	
AD0 (LSB)		I/O	
CCLK		I	- Character Clock - this establishes the basic character clock rate and is used to synchronize the controller internal timing.
CSL3 (MSB)		O	- Character Scan Line - CSL3 through CSL0 indicate the binary number of the scan line for the character to be displayed.
CSL2		O	
CSL1		O	
CSL0 (LSB)		O	
CURSOR		O	- Cursor - a positive true signal indicating the location in the Display Memory defined to have the cursor.
UNPOS		O	- Underline Position - a positive true signal indicates that the underline attribute should be displayed if applicable.
BLINK		O	- Blink rate - this signal determines the rate at which characters should be blinked when applicable.
$\overline{\text{DMWR}}$		O	- Write Display Memory - A negative true signal indicates that the data gated onto the character and attribute data bus will be stored in the Display Memory at the address provided by the Display Memory Address Bus.
CGA		O	- Character Generator Access - a positive true signal indicates that the controller is accessing the Character Generator Circuitry instead of the Display Memory.
ADF		O	- Alternate Dsiplay Format - a positive true signal indicates that the controller has been configured to operate in the Alternate Display Format instead of the standard format.
$\overline{\text{COUT}}$		O	- Control Output - this output signal is the inverse of Bit 6 of the Control Register of the controller.

KEYBOARD AND KEYBOARD INTERFACE UNIT

Keyboard unit 58 is self-contained, with its own microcomputer 64, for scanning and decoding the keys, assigning various function thereto and transmitting and receiving data and instructions serially via I/F 60 to CPU 10. Various character formats are selectable, e.g., for different countries as stored in character RAM 106. The advantages of having separate keyboard unit include a reduced processing load on CPU 10 and thus a faster throughput.

Referring now to Figure 9, keyboard unit 58 includes an array or matrix 166 of momentarily-closed single action switches arranged in rows and columns. Microcomputer 64, for example, a model 3870 available from MOSTEK Corporation and from Motorola, Inc., includes internal ROM and RAM. Input/output ports, P0 and P4 are connected to a series of open collector buffers U2-U4, used for strobing matrix 166. A feedback return is produced via a row path into port P5 of U1. Four control keys, shown in the left hand portion of Figure 9, are connected to U1 through input port P1. Bits 5 and 6 of port P0 are connected to a bidirectional data line E3 and a clock line E4 which are connected in turn to keyboard I/F 60 for serial transmission of signals thereto and therefrom. Lines E1 and E2 provide power and ground connection for keyboard unit 58. The for logic gates denoted as U5 in Figure 9 form a bell or tone circuit that is controlled by microcomputer 64 upon commands from CPU 10 and from internal error codes.

Referring now to Figure 10, I/F 60 takes serialized data from keyboard data 58 and converts the data to a parallel form in a tri-state output shift register 168 for output onto I/O data bus 46. A series of flip-flop circuits FF1-FF4 control the direction of data to shift register 168. FF1 determines whether I/F 60 is in a read or a write mode and controls the data path. FF2 signals microcomputer 64 that CPU 10 is still processing data and is not ready to read yet. FF3 sends an interrupt signal to CPU 10 to indicate that the data in register 168 is available for reading, and FF4 establishes a time-delay to stabilize the data transfer into register 168 from the I/O data bus 46. The signals that appear on the left

hand portion of Figure 10 are from CPU 10. As stated above, this arrangement provides a reduced load on system processor CPU 10, while expanding the versatility of keyboard unit 58.

Figure 10a is a timing diagram showing the relationship of various signals in I/F 60 to the clock and data signals. The clock signal is generated by microcomputer 64, and the serial data in the read and the write modes is bi-directional.

PREFERRED MODE OF OPERATION

Appendix A forms a map, in hexadecimal code, of the contents of the GROM program storage memory 52. These contents are instructions that cause the present system to operate in a prescribed way. The instructions include device service systems, application programs, protocols, command processors and the like. Such instruction lists are known for causing perpherial devices to operate in a desired manner. These instructions are sent to the DRAM memory 48 via system microprocessor CPU 10 shown in Figure 1. Similarly, Appendix B forms a map of the contents of CMOS RAM 30 which, when connected to battery 31, provides nonvolatile terminal configuration information which is used by the system after a power off condition. Appendix C forms a map of the contents of system ROM 27, which contains the system initialization instructions.

The program instructions stored in memories 27, 30, and 52 were written specifically for a Zilog Corporation Model Z-80 microprocessor (CPU 10). However, similar microprocessors will work equally as well in the disclosed system. CTC 18, DART 20 and SIO 22 are also available from the Zilog Corporation and are compatible with the present system. Upon power up, CPU 10 first initiates self-tests on each of the ROM memories. Then the operating system DRAM 48 is loaded in a predetermined sequence from the GROM 52. The system is then ready for either to accept a message from the host computer or to perform a task under local operator control.

Thus, there is provided by the present invention an electronic data system including means for segmenting a display screen into a plurality of independently controllable vertical and horizontally split regions. Smooth or discrete scrolling operations may be performed in any selected horizontal or vertical split region. The display controller system, which operates asychronously with the system microprocessor provides access to the display memory character ROM and character RAM as input/output devices rather than as directly connected memory. The display controller accesses and modifies data in display memory by means of line attributes or pointers associated with each row of characters therein. This system permits data to be

rearranged in display memory and on the display screen on an individual region basis without the transfer of whole blocks of data, which is required in systems using software for display control.

Other embodiments and modifications of the present invention will readily come to those of ordinary skill in the art having the benefit of the teachings presented in the foregoing description and drawings. It is therefore to be understood that this invention is not to be limited thereto and that said modification and embodiments are intended to be included within the scope of the appended claims.

What is claimed is:

1. A data display system for forming a plurality of independently controllable display regions on a display device comprising:

data processor means;

main memory means connected to said data processor means;

input means connected to said data processor means and to said main memory means;

display controller means connected to said data processor means;

display memory means connected to said display controller means wherein each row of characters to be displayed on said display device is assigned one or more memory location identifiers as defined by said display controller means to indicate the display memory address of a subsequent row of characters to be displayed and wherein said display controller means alters the sequence or rows displayed on said display device by modifying the memory location identifiers of the selected rows of characters.

2. The system of Claim 1, further comprising :

character memory means connected to said display controller means for storing a predetermined set of character data codes, whereby the data code corresponding to a selected character is loaded into said display memory means in response to a command from said display controller means.

3. The system of Claim 2, further comprising :

generator means connected to said display controller means, to said character memory means and to said display memory means for outputting control signals to

said display device and for generating a visual attribute code for each character stored in said display memory means; and

character random access memory means connected to said display controller means and to said generator means for storing alterable character data codes therein, wherein a selected character code is transferred therefrom through said generator means into said display memory means in response to a command from said display controller means.

4. The system of Claim 3 wherein said display device comprises a chathode-ray tube, further comprising:

means connected to said data processor means and to said generator means for controlling the contrast of said cathode-ray tube in a plurality of discrete steps.

5. The system of Claim 4, wherein said data processor means comprises a microprocessor.

6. The system of Claim 5, wherein said main memory means comprises a random access memory.

7. The system of Claim 6, wherein said input means comprises:

an array of manually operable switch means;

second data processor means for scanning said array of switch means wherein said second data processor means transmits electronic signals to said microprocessor corresponding to which of said switch means are enabled.

8. A display control system comprising:

character memory means;

display memory means connected to said character memory means;

visual attribute generator means connected to

said display memory means and said character memory means;

means for loading character and visual attribute data into selected display memory locations from a data bus, wherein said character data is stored in said character memory means and said attribute data is generated by said visual attribute generator means, and wherein a plurality of character and attribute data bytes form a block data in the display memory corresponding to a row of characters as seen on the display device;

means for assigning one or more memory location identifier bytes to each displayed row of characters, corresponding to said data blocks in the display memory, wherein each of said identifier bytes contains the display memory address of the first character in the subsequent row of characters to be displayed on the display device;

means for storing a data byte corresponding to a preselected column on said display device wherein a vertical split region will begin;

second means for storing a data byte corresponding to the starting address of a character row;

third means for storing a data byte corresponding to the starting row address of a vertical split region; and

means for determining which scan line of a character a particular display memory address corresponds to, whereby a plurality of rows of characters appearing on the display device are scrolled one scan line at a time.

9. The display controller device of Claim 8, further including:

means for assigning within said memory location identifier bytes selected region row visual attribute information: and

means for presenting the next row of characters on said display device in a selected manner corresponding to said selected visual attributes.

10. The display controller device of Claim 9 wherein said selected region row visual attribute comprises double high characters, whereby each scan line of region character row information is repeated on said display device before the next character row scan line is accessed from said display memory.

11. A method for presenting data on a display device in a plurality of independently controllable segmented regions comprising the steps of:

generating and assigning unique row identifier codes to each row of data in each of said regions, wherein each of said row identifier codes indicates a location in a display memory;

interpreting said region row identifier codes;

accessing the display memory location identified by a particular row identifier code;

presenting the next row of data on said display device and

repeating said steps of interpreting, accessing and presenting for each frame of data presented on said display device.

12. The method of Claim 11 wherein said region row identifier codes contain preselected visual attribute information for each row of data, further including the steps of:

interpreting said visual row attribute information included in said row identifier codes before accessing said display memory; and

presenting the next row of data in the manner indicated by preselected row attributes.

13. The method of Claim 12 wherein each row of data is formed of a predetermined number of scan lines on said display device and a smooth scroll attribute is selected

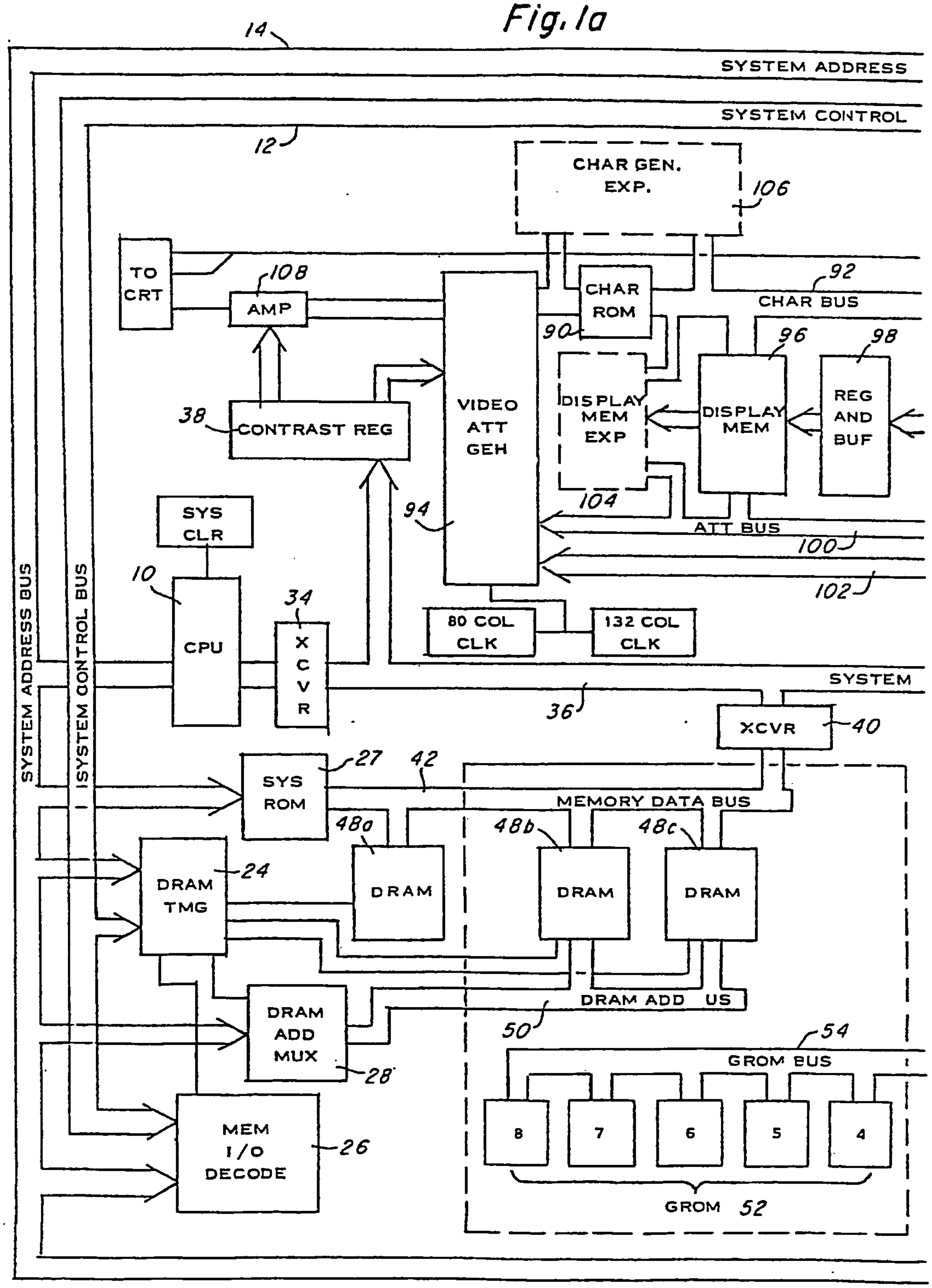
for a particular region, further comprising the step of:

offsetting the data in said particular region by one scan line for each frame of data presented on said display device until said smooth scroll is completed.

14. The method of Claim 13 wherein a double high character attribute is selected for a particular region row, further comprising the step of:

repeating the presentation of each scan line of information for said selected region row on said display device before accessing the next region character row scan line from said display memory.

Fig. 1a


14
SYSTEM ADDRESS
SYSTEM CONTROL
12
CHAR GEN. EXP.
106
TO CRT
108
AMP
CHAR ROM
92
CHAR BUS
90
96
98
38
CONTRAST REG
VIDEO ATT GEN
DISPLAY MEM EXP
DISPLAY MEM
REG AND BUF
104
94
ATT BUS
100
102
SYS CLK
10
CPU
34
XCVR
80 COL CLK
132 COL CLK
SYSTEM ADDRESS BUS
SYSTEM CONTROL BUS
SYSTEM
36
XCVR
40
27
42
SYS ROM
MEMORY DATA BUS
48a
48b
48c
DRAM TMG
24
DRAM
DRAM
DRAM
DRAM ADD BUS
50
DRAM ADD MUX
28
54
GROM BUS
MEM I/O DECODE
26
8
7
6
5
4
GROM 52

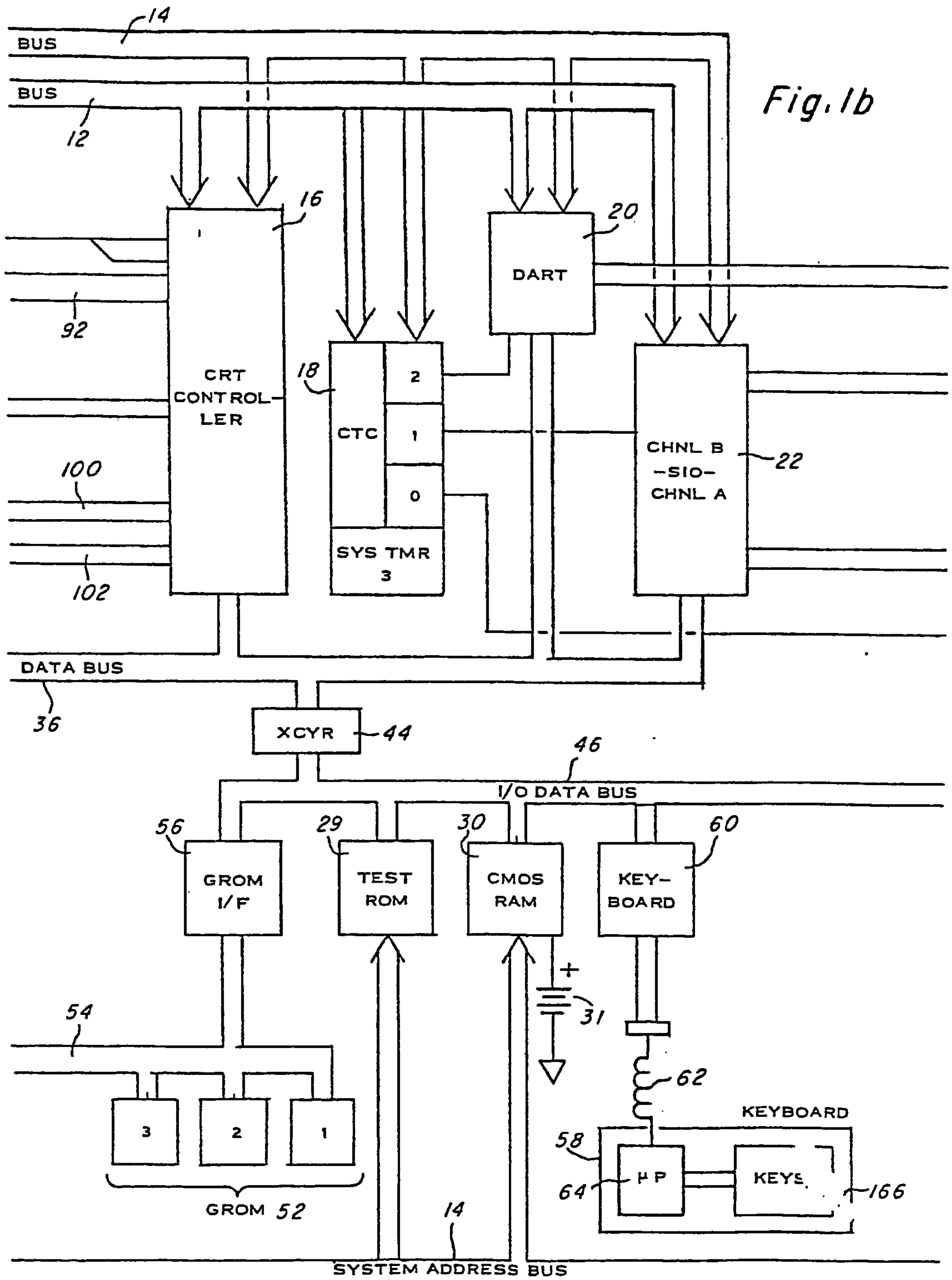
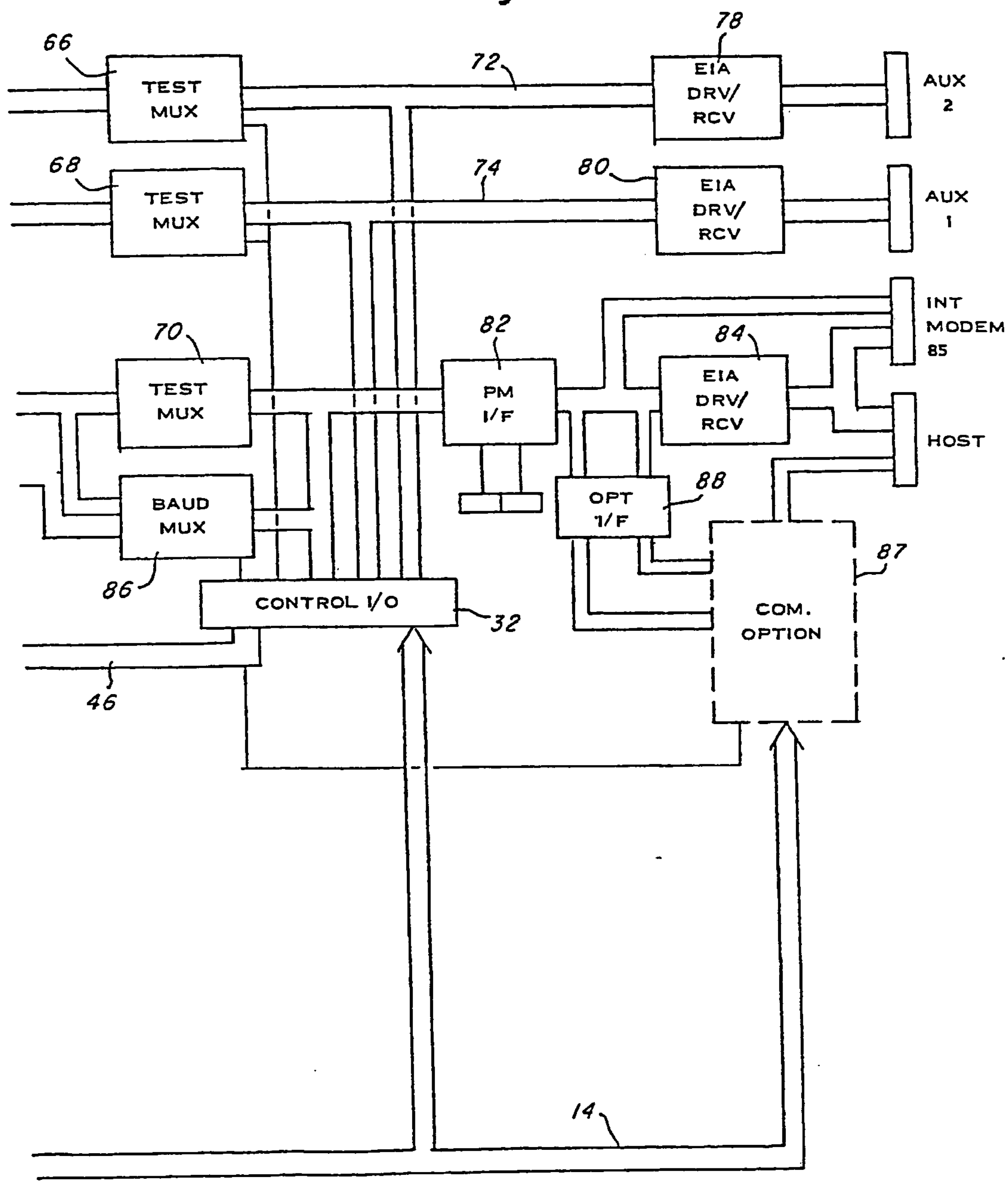

14
BUS
BUS
12
16
20
DART
92
CRT
CONTROL-
LER
18
CTC
2
1
0
SYS TMR
3
CHNL B
-SIO-
CHNL A
22
100
102
DATA BUS
36
XCYR
44
46
I/O DATA BUS
56
29
30
60
GROM
I/F
TEST
ROM
CMOS
RAM
KEY-
BOARD
+
31
54
62
KEYBOARD
58
3
2
1
μP
KEYS
64
166
GROM 52
14
SYSTEM ADDRESS BUS

Fig. 1b

Fig. 1c


66
TEST MUX
68
TEST MUX
70
TEST MUX
BAUD MUX
86
72
74
78
EIA DRV/ RCV
80
EIA DRV/ RCV
AUX 2
AUX 1
82
PM I/F
84
EIA DRV/ RCV
INT MODEM 85
HOST
OPT I/F
88
87
COM. OPTION
CONTROL I/O
32
46
14

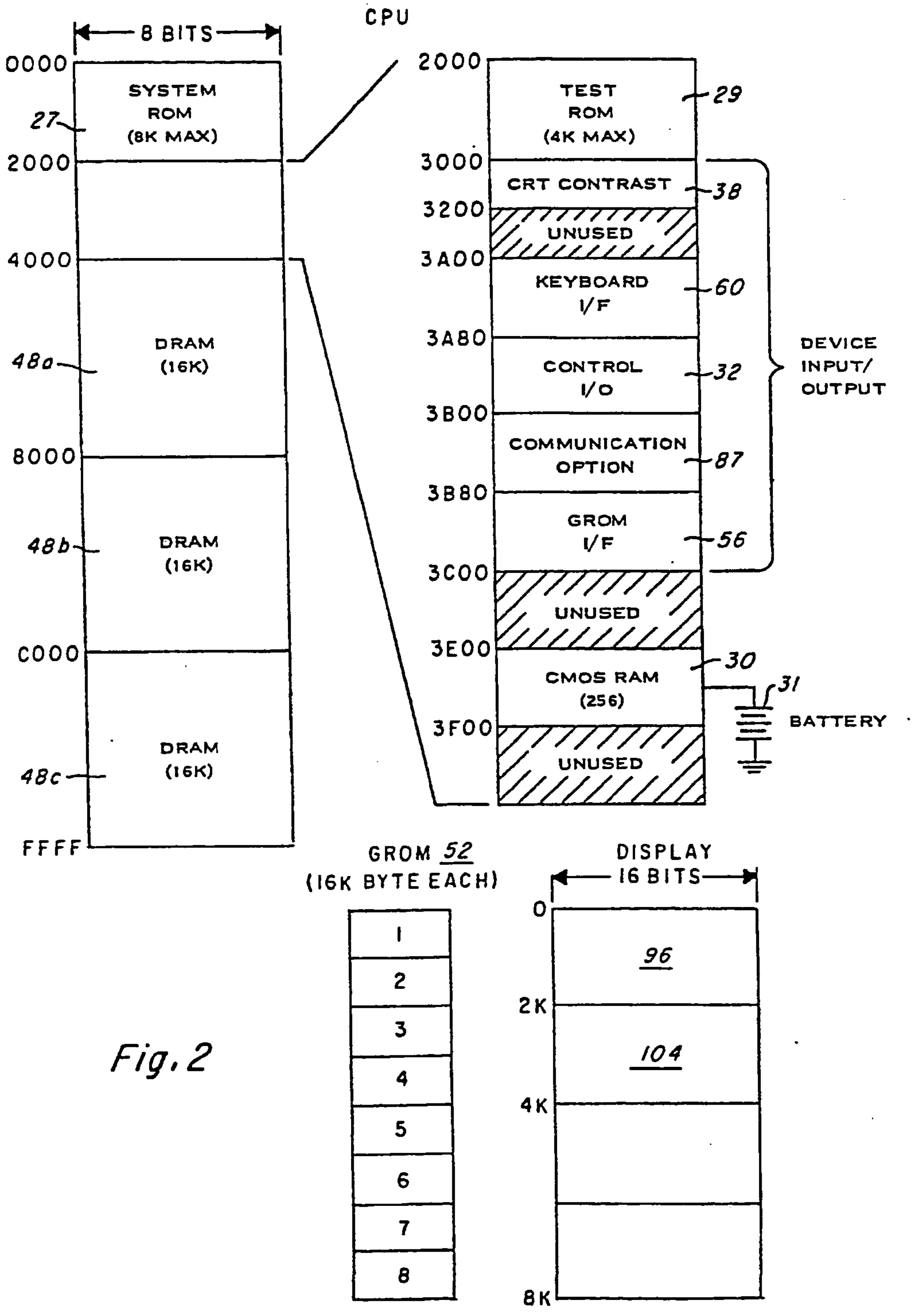

8 BITS
CPU
0000
SYSTEM ROM (8K MAX)
27
2000
4000
DRAM (16K)
48a
8000
DRAM (16K)
48b
C000
DRAM (16K)
48c
FFFF
2000
TEST ROM (4K MAX)
29
3000
CRT CONTRAST
38
3200
UNUSED
3A00
KEYBOARD I/F
60
3A80
CONTROL I/O
32
3B00
COMMUNICATION OPTION
87
3B80
GROM I/F
56
3C00
UNUSED
3E00
CMOS RAM (256)
30
3F00
UNUSED
31
BATTERY
DEVICE INPUT/ OUTPUT
GROM 52 (16K BYTE EACH)
1
2
3
4
5
6
7
8
DISPLAY 16 BITS
0
96
2K
104
4K
8K

Fig. 2

Fig. 3

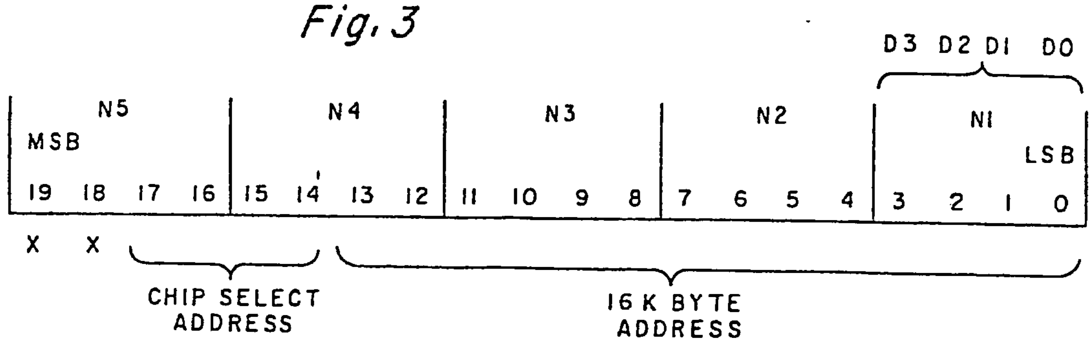
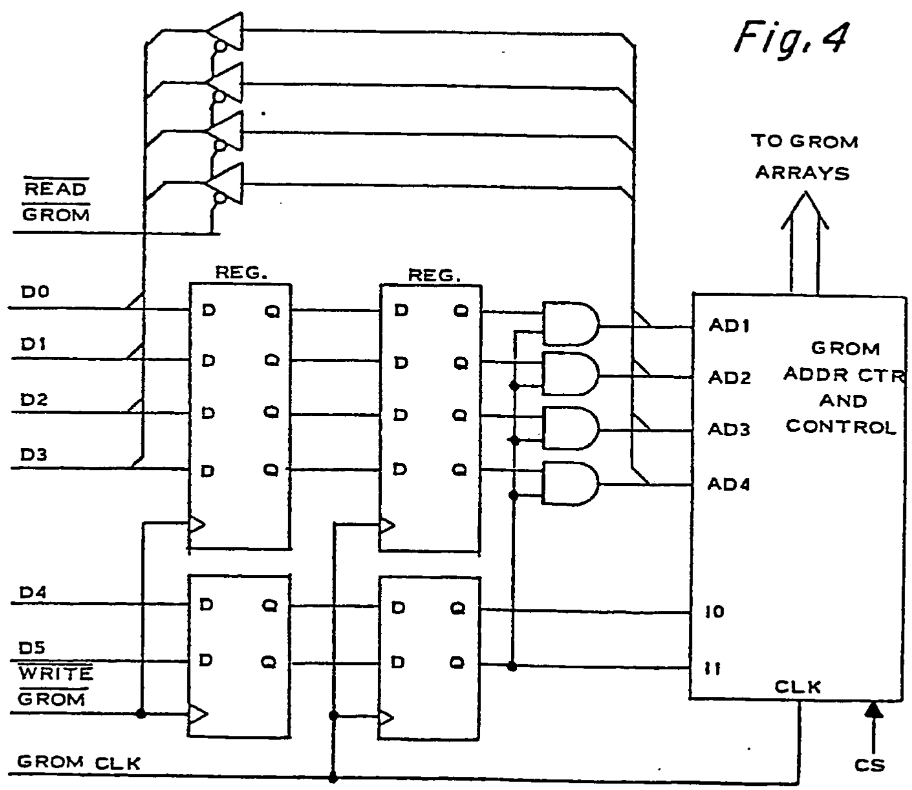

D3 D2 D1 D0
N5
MSB
N4
N3
N2
N1
LSB
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
X X
CHIP SELECT
ADDRESS
16 K BYTE
ADDRESS

Fig. 4


TO GROM
ARRAYS
READ
GROM
REG.
REG.
D0
D1
D2
D3
D4
D5
WRITE
GROM
GROM CLK
D Q
AD1
AD2
AD3
AD4
GROM
ADDR CTR
AND
CONTROL
I0
I1
CLK
CS

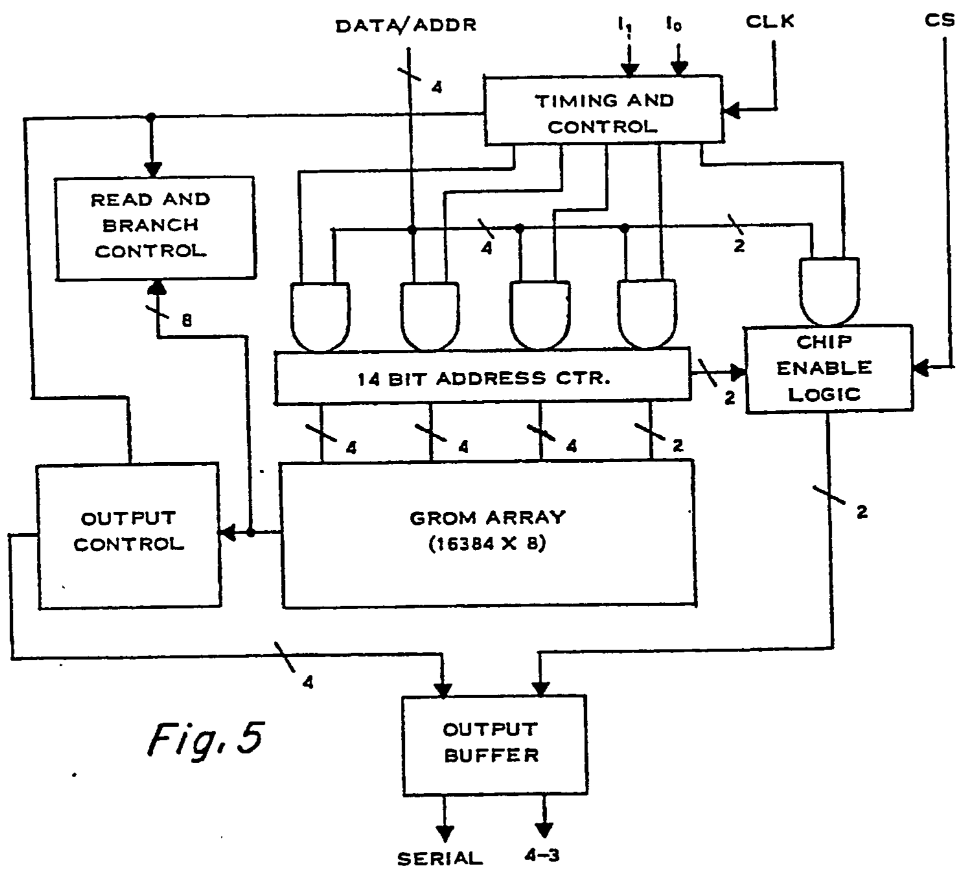
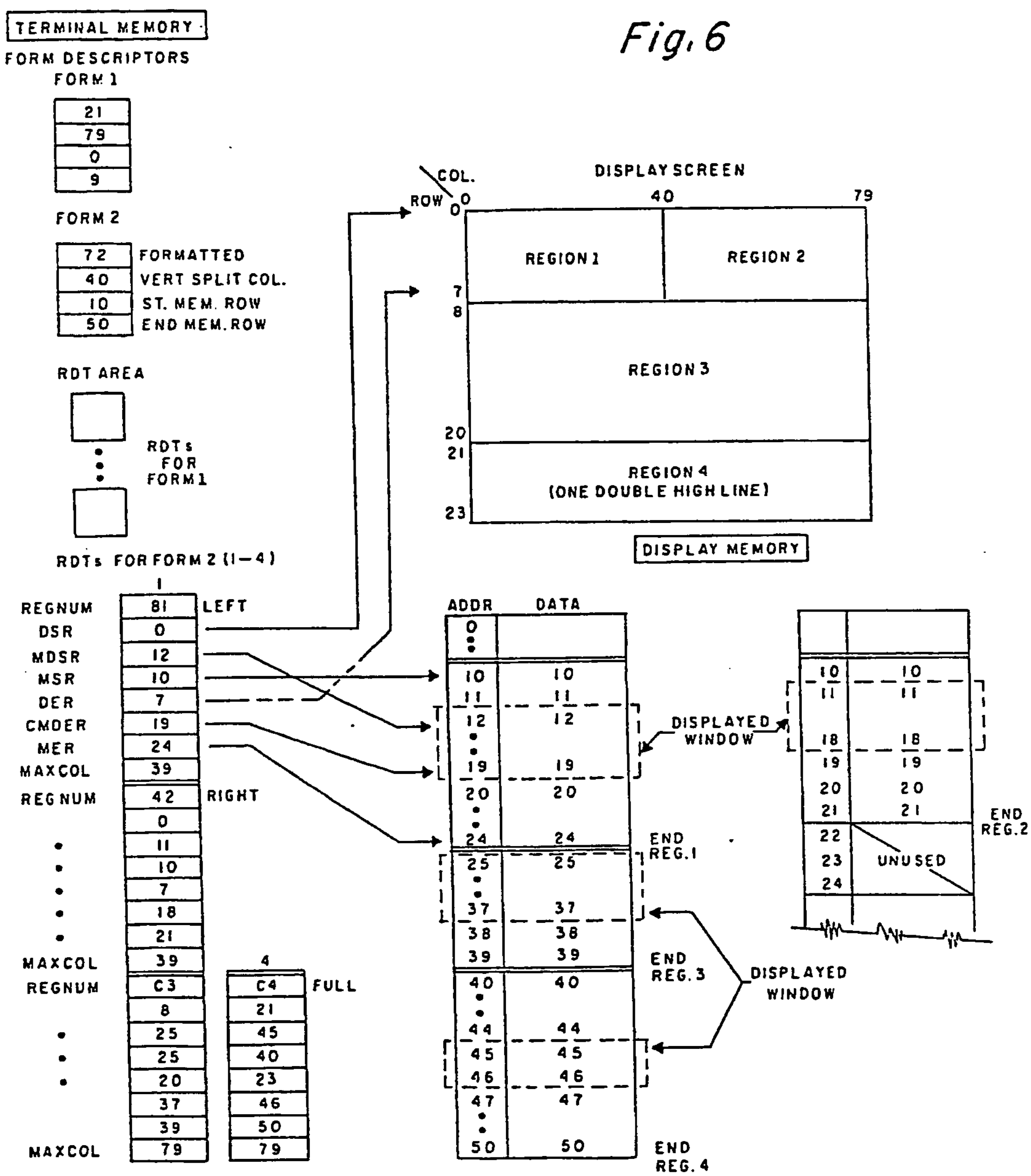

DATA/ADDR
I1
I0
CLK
CS
4
TIMING AND CONTROL
READ AND BRANCH CONTROL
4
2
8
14 BIT ADDRESS CTR.
2
CHIP ENABLE LOGIC
4
4
4
2
2
OUTPUT CONTROL
GROM ARRAY (16384 X 8)
4
OUTPUT BUFFER
SERIAL
4-3

Fig. 5

Fig. 6


TERMINAL MEMORY
FORM DESCRIPTORS
FORM 1
21
79
0
9
FORM 2
72 FORMATTED
40 VERT SPLIT COL.
10 ST. MEM. ROW
50 END MEM. ROW
RDT AREA
RDTs FOR FORM 1
RDTs FOR FORM 2 (1–4)
REGNUM
DSR
MDSR
MSR
DER
CMDER
MER
MAXCOL
LEFT
RIGHT
FULL
DISPLAY SCREEN
COL.
ROW
REGION 1
REGION 2
REGION 3
REGION 4
(ONE DOUBLE HIGH LINE)
DISPLAY MEMORY
ADDR
DATA
DISPLAYED WINDOW
END REG. 1
END REG. 2
END REG. 3
END REG. 4
UNUSED

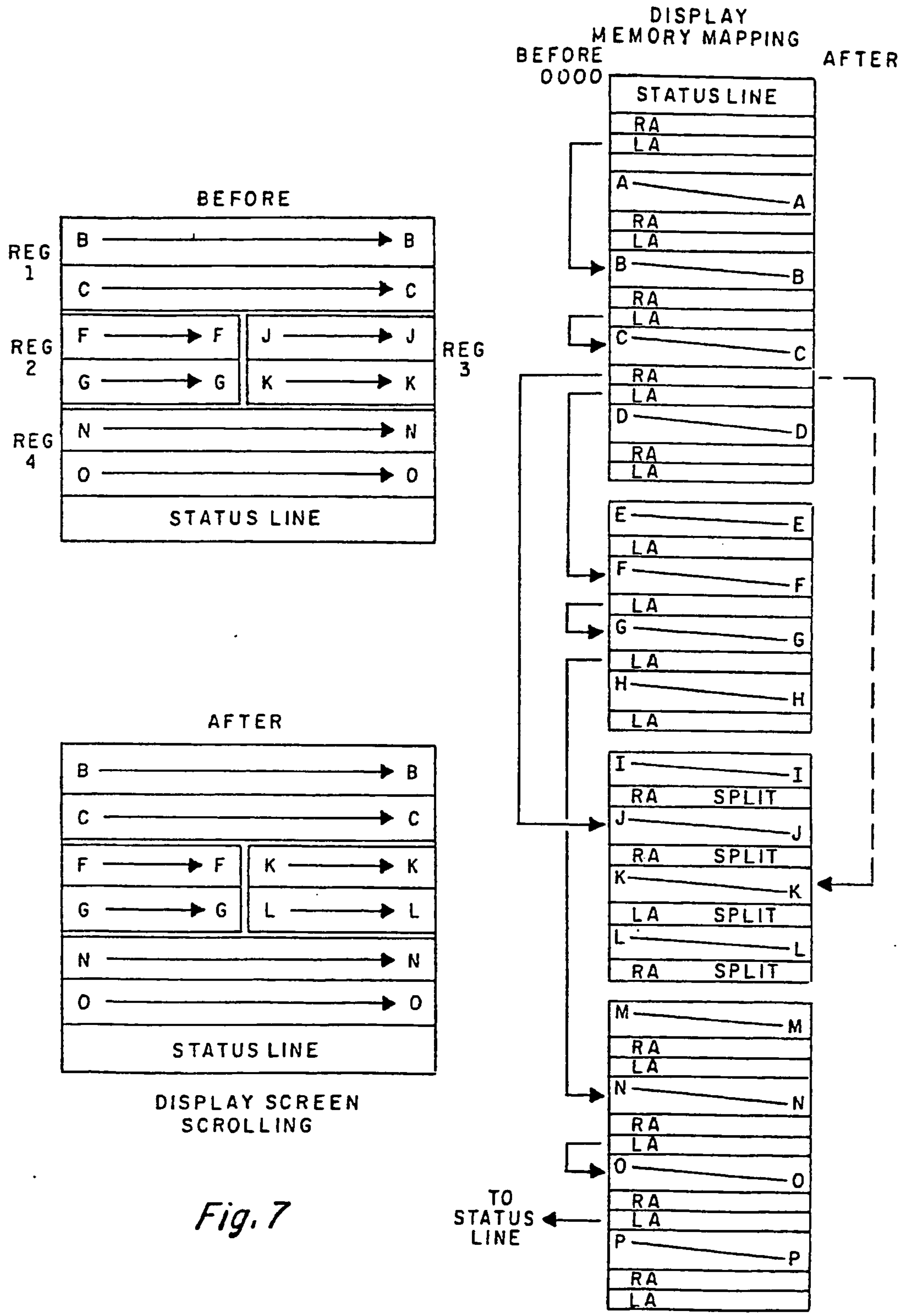

DISPLAY MEMORY MAPPING
BEFORE
0000
AFTER
STATUS LINE
RA
LA
A
B
C
D
E
F
G
H
I
J
K
L
SPLIT
M
N
O
P
TO STATUS LINE
BEFORE
REG 1
REG 2
REG 3
REG 4
STATUS LINE
AFTER
DISPLAY SCREEN SCROLLING

Fig. 7

Fig. 8a

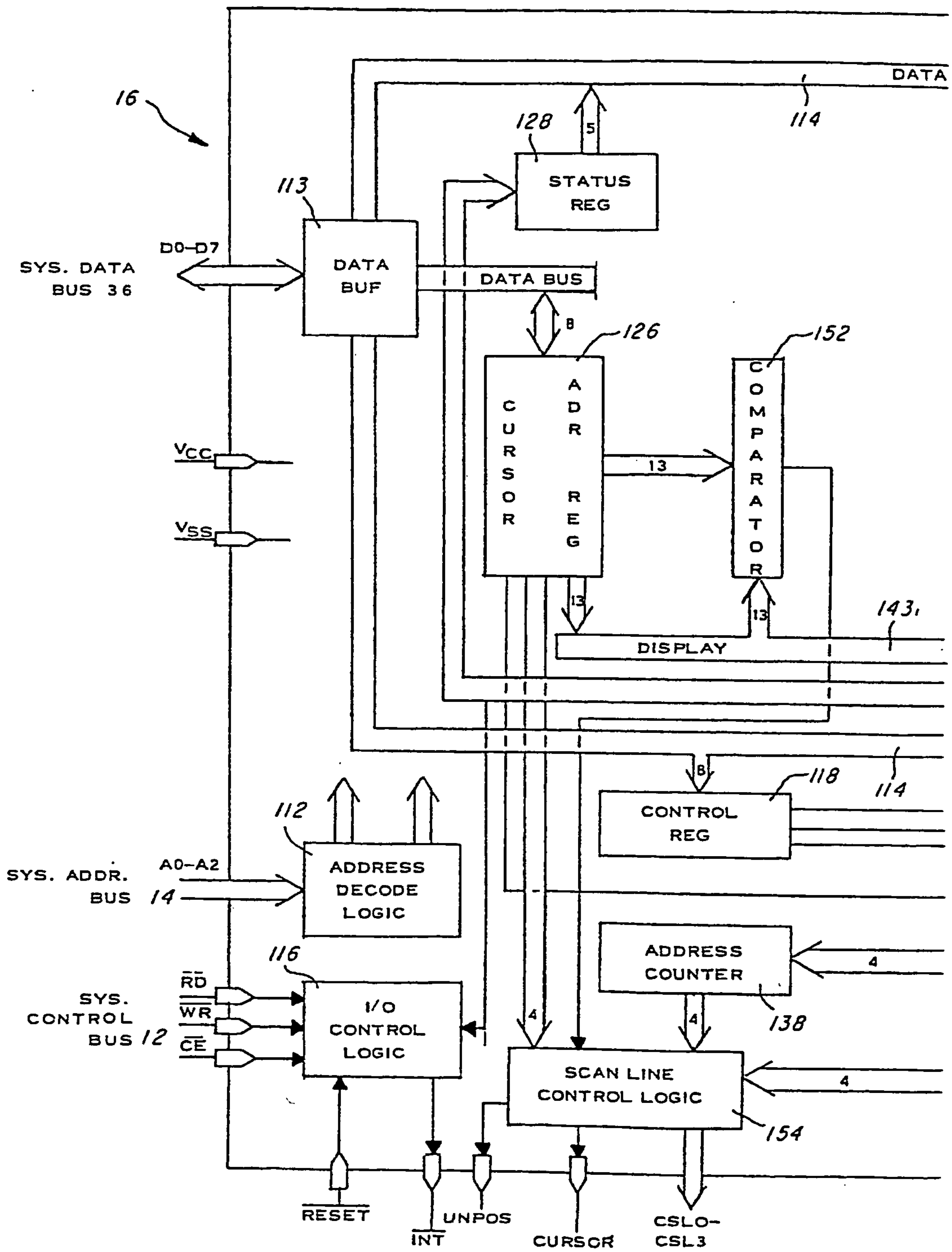

16
DATA
114
128
5
STATUS REG
113
D0–D7
SYS. DATA BUS 36
DATA BUF
DATA BUS
8
126
152
CURSOR ADR REG
COMPARATOR
13
Vcc
Vss
13
13
143
DISPLAY
8
118
114
CONTROL REG
112
ADDRESS DECODE LOGIC
A0–A2
SYS. ADDR. BUS 14
ADDRESS COUNTER
4
138
116
$\overline{RD}$
$\overline{WR}$
$\overline{CE}$
SYS. CONTROL BUS 12
I/O CONTROL LOGIC
4
4
SCAN LINE CONTROL LOGIC
4
154
$\overline{RESET}$
$\overline{INT}$
UNPOS
CURSOR
CSL0–CSL3

Fig. 8b

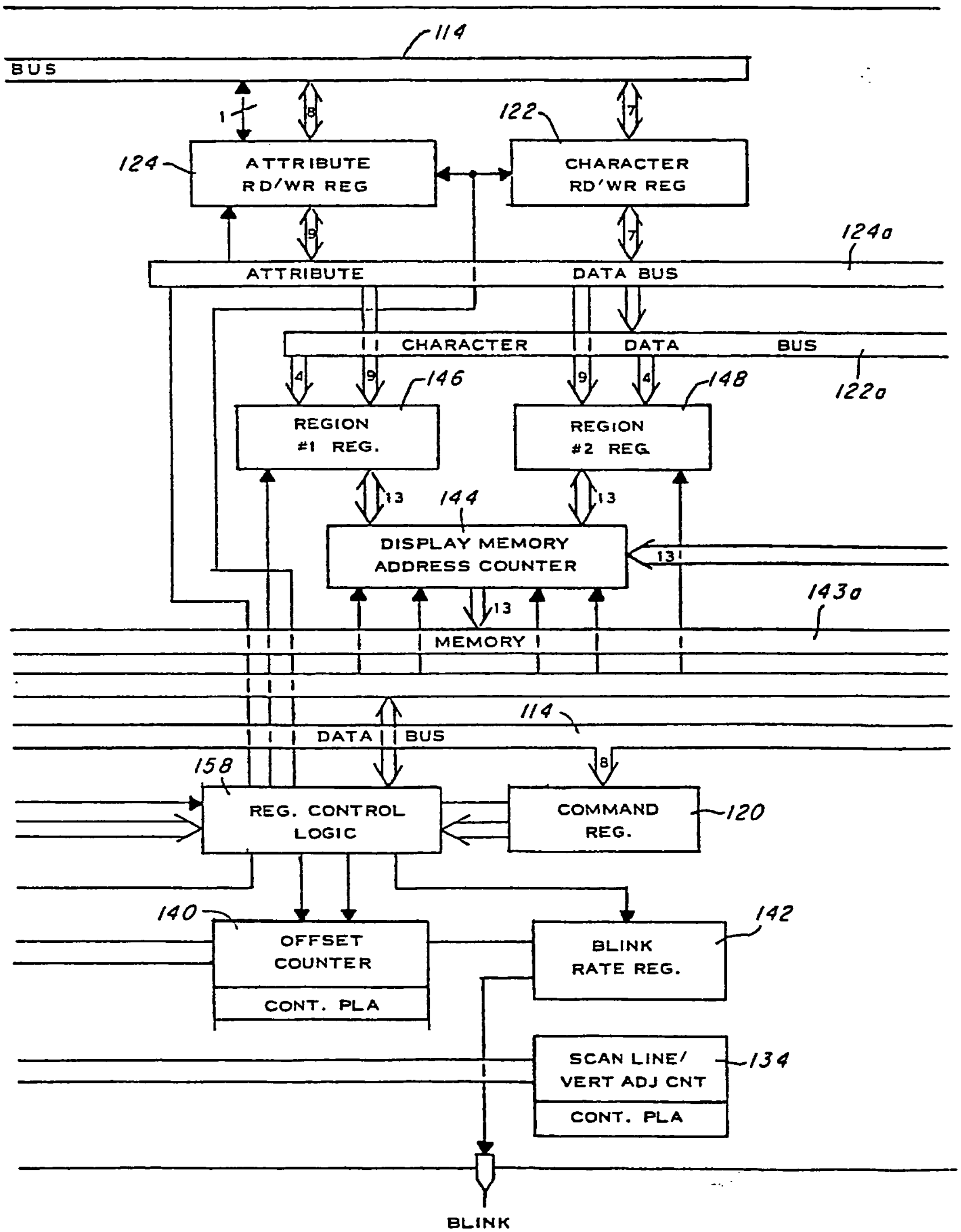
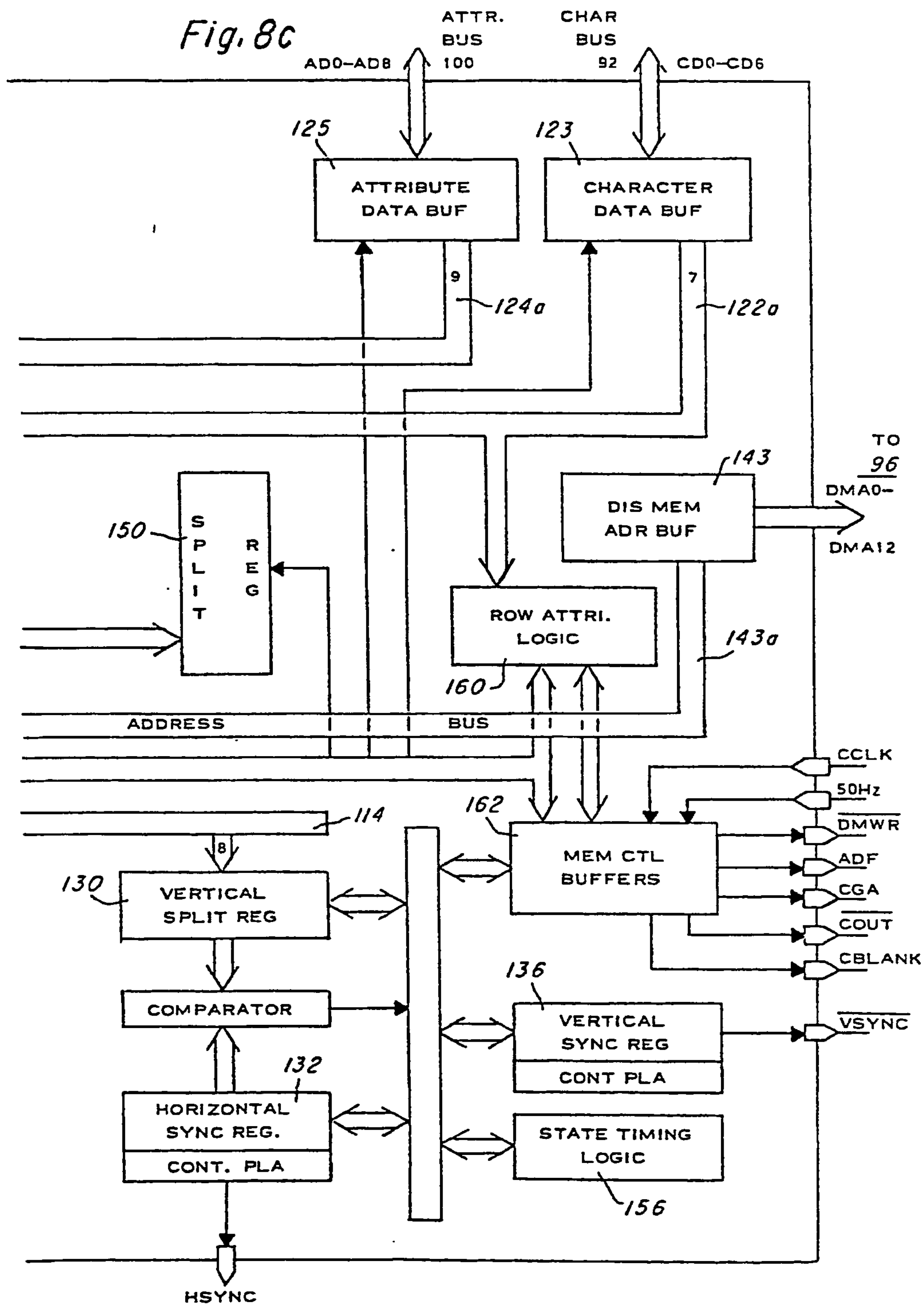

114
BUS
1
8
7
122
124
ATTRIBUTE
RD/WR REG
CHARACTER
RD'WR REG
8
7
124a
ATTRIBUTE
DATA BUS
CHARACTER
DATA
BUS
122a
4
9
146
9
4
148
REGION
#1 REG.
REGION
#2 REG.
13
144
13
DISPLAY MEMORY
ADDRESS COUNTER
13
143a
13
MEMORY
114
DATA
BUS
8
158
REG. CONTROL
LOGIC
COMMAND
REG.
120
140
OFFSET
COUNTER
CONT. PLA
BLINK
RATE REG.
142
SCAN LINE/
VERT ADJ CNT
CONT. PLA
134
BLINK

Fig. 8c


ATTR. BUS 100
AD0–AD8
CHAR BUS 92
CD0–CD6
125
ATTRIBUTE DATA BUF
123
CHARACTER DATA BUF
9
124a
7
122a
143
DIS MEM ADR BUF
TO 96
DMA0–
DMA12
150
SPLIT REG
ROW ATTRI. LOGIC
160
143a
ADDRESS
BUS
CCLK
50Hz
DMWR
ADF
CGA
COUT
CBLANK
114
162
8
130
VERTICAL SPLIT REG
MEM CTL BUFFERS
COMPARATOR
136
VERTICAL SYNC REG
CONT PLA
VSYNC
132
HORIZONTAL SYNC REG.
CONT. PLA
STATE TIMING LOGIC
156
HSYNC

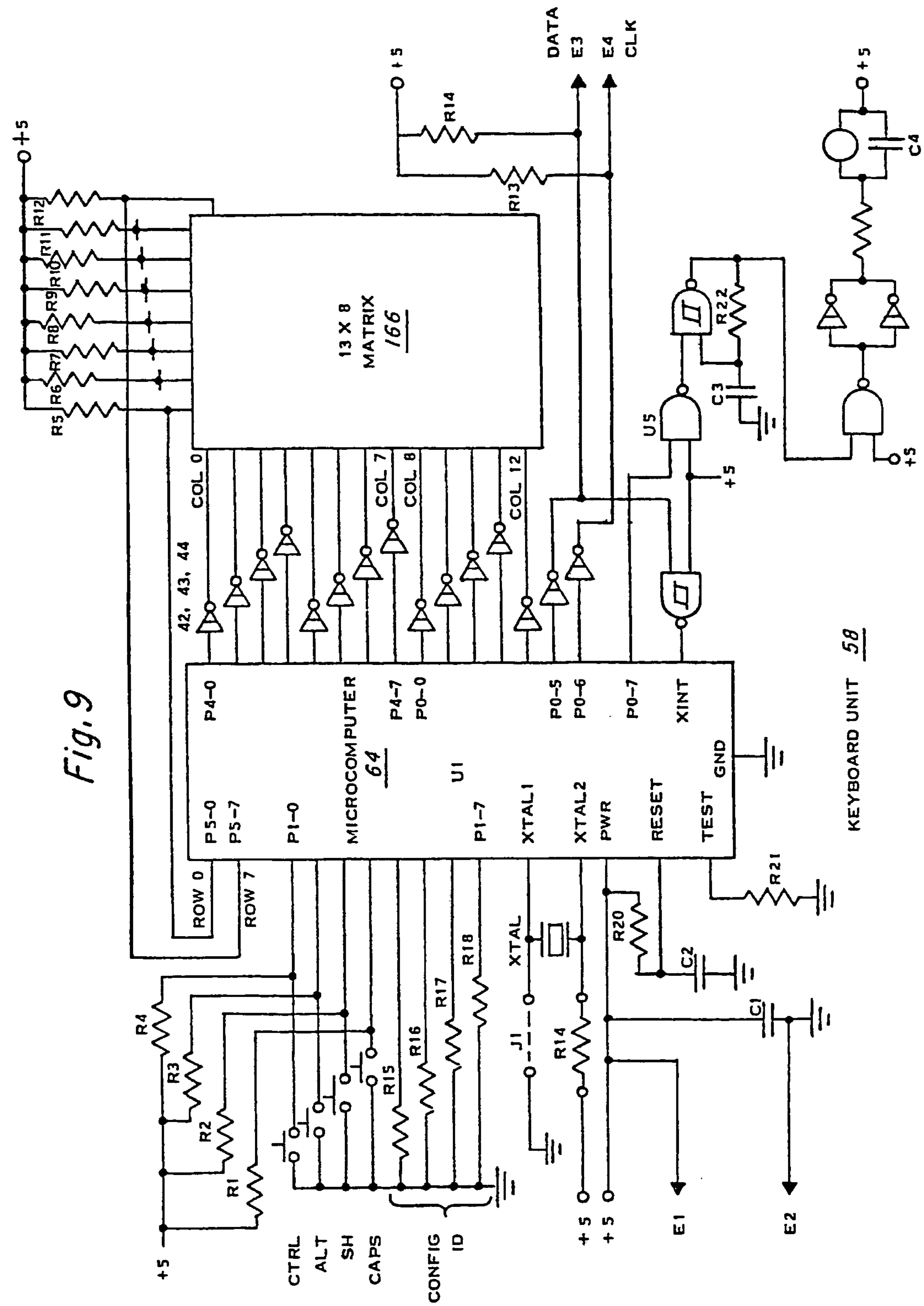

13 X 8
MATRIX
166
MICROCOMPUTER
64
U1
KEYBOARD UNIT 58
DATA
E3
E4
CLK
COL 0
COL 7
COL 8
COL 12
42, 43, 44
ROW 0
ROW 7
P4-0
P4-7
P0-0
P0-5
P0-6
P0-7
XINT
GND
TEST
RESET
PWR
XTAL2
XTAL1
P1-7
P1-0
P5-7
P5-0
XTAL
J1
U5
CTRL
ALT
SH
CAPS
CONFIG
ID
E1
E2
+5

Fig. 9

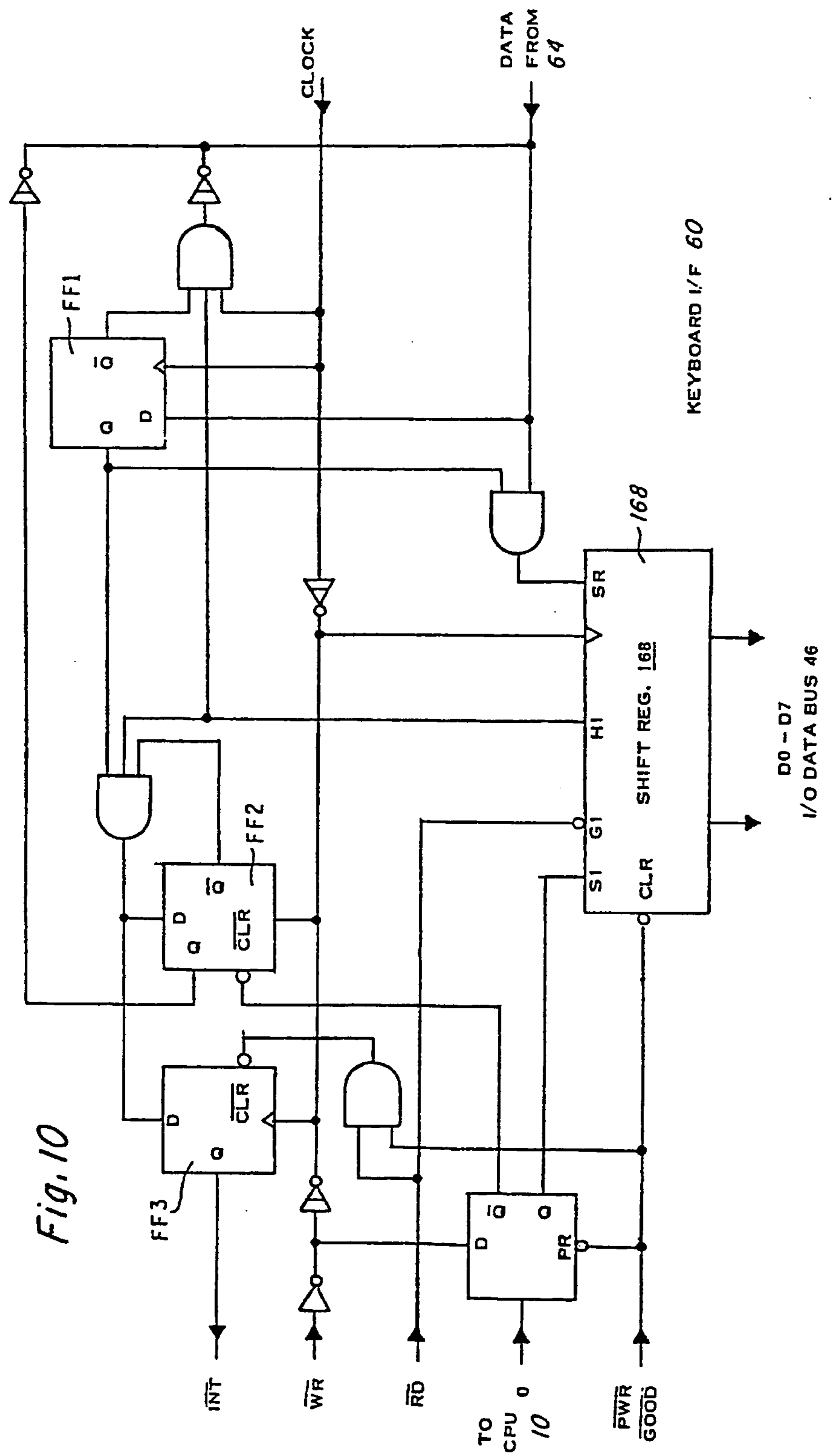
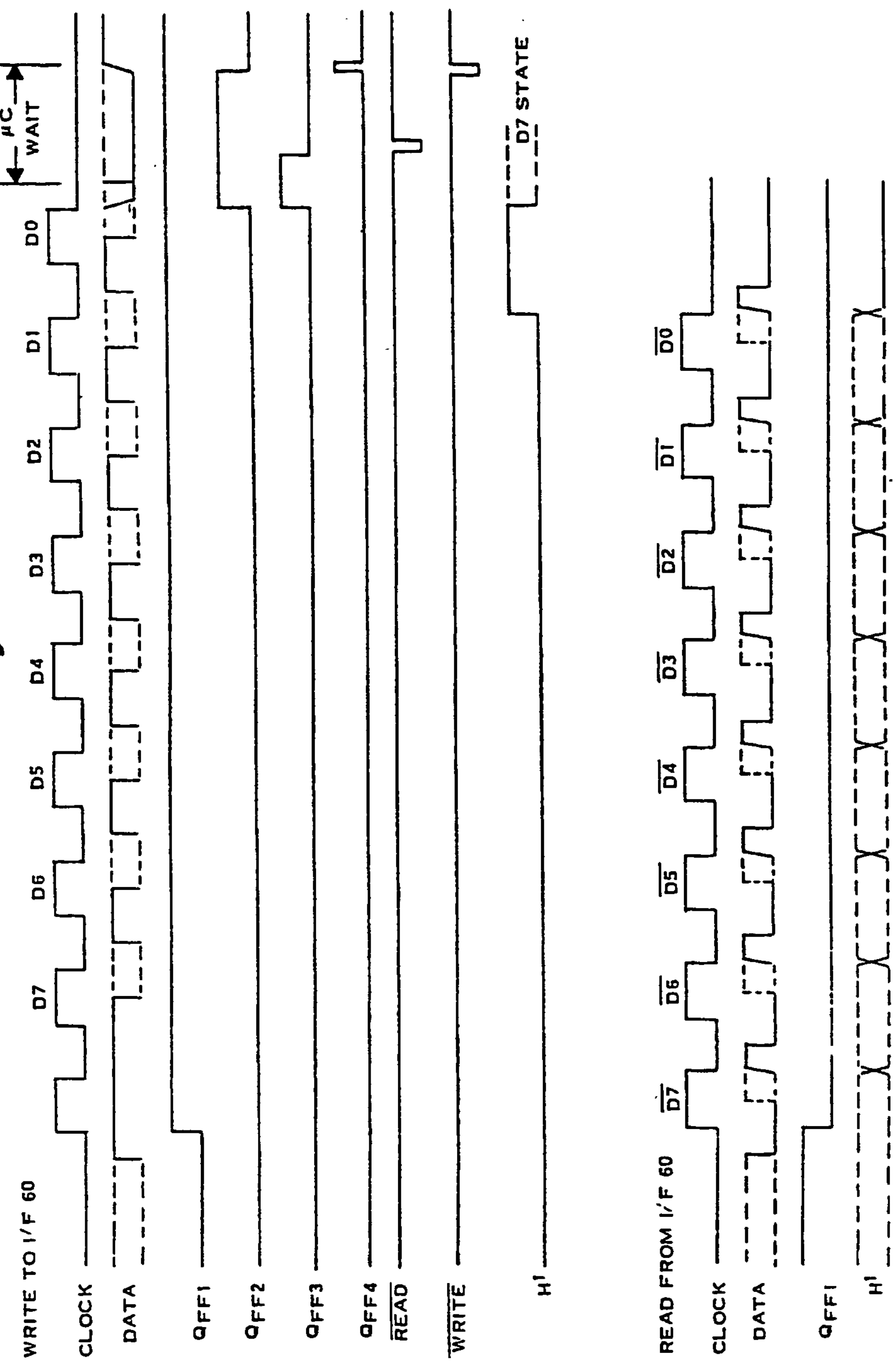

CLOCK
DATA FROM 64
KEYBOARD I/F 60
FF1
FF2
FF3
168
SHIFT REG. 168
SR
HI
GI
SI
CLR
D0 - D7
I/O DATA BUS 46
INT
WR
RD
TO CPU 10
PWR GOOD

Fig. 10

Fig. 10a


WRITE TO I/F 60
CLOCK
DATA
QFF1
QFF2
QFF3
QFF4
READ
WRITE
H'
D7
D6
D5
D4
D3
D2
D1
D0
μC WAIT
D7 STATE
READ FROM I/F 60
CLOCK
DATA
QFF1
H'
D7
D6
D5
D4
D3
D2
D1
D0